

Requested Patent WO9706631A2
Title: APPARATUS AND METHOD FOR OBJECT TRACKING ;
Abstracted Patent: WO9706631 ;
Publication Date: 1997-02-20 ;
Inventor(s): PASTOR YOSEF (IL); SPIEGEL EHUD (IL) ;
Applicant(s): PASTOR YOSEF (IL); SPIEGEL EHUD (IL) ;
Application Number: WO1996IL00070 19960802 ;
Priority Number(s): IL19950114839 19950804 ;
IPC Classification: H04N ;
Equivalents: AU6530396, CA2228619, EP0880852 (WO9706631), A3, JP11510351T

ABSTRACT:

A tracking method including receiving a representation of an event including at least one dynamic object having a border and having at least one edge portion which is absent during at least a portion of the event, and providing an ongoing indication of the location of the border of the object during the event.



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ :	A2	(11) International Publication Number: WO 97/06631
H04N		(43) International Publication Date: 20 February 1997 (20.02.97)
		(21) International Application Number: PCT/IL96/00070
		(22) International Filing Date: 2 August 1996 (02.08.96)
(30) Priority Data:		114839 4 August 1995 (04.08.95) IL
(71)(72) Applicants and Inventors: SPIEGEL, Ehud [IL/IL]; 10 Mordei Hagetaot Street, 76464 Rehovot (IL). PASTOR, Yosef [IL/IL]; 56 Hagalil Street, 32225 Haifa (IL).		
(74) Agents: SANFORD, T., Colb et al.; Sanford T. Colb & Co., P.O. Box 2273, 76122 Rehovot (IL).		
(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).		
<p>Published <i>Without international search report and to be republished upon receipt of that report.</i></p>		
(54) Title: APPARATUS AND METHOD FOR OBJECT TRACKING		
<p>(57) Abstract</p> <p>A tracking method including receiving a representation of an event including at least one dynamic object having a border and having at least one edge portion which is absent during at least a portion of the event, and providing an ongoing indication of the location of the border of the object during the event.</p>		
<pre> graph TD VIDEOINTERFACE[VIDEO INTERFACE] -- "VIDEO ARRIVING OVER A NETWORK" --> CAM[VIDEO CAMERA] VIDEOINTERFACE -- "VIDEO ARRIVING OVER A NETWORK" --> STORAGE[VIDEO STORAGE UNIT] VIDEOINTERFACE -- "VIDEO ARRIVING OVER A NETWORK" --> FILMSCANNER[FILM SCANNER] PROCESSINGUNIT[PROCESSING UNIT] <--> VIDEOMEMORY[VIDEO MEMORY] PROCESSINGUNIT <--> VIDEOCOMPOSINGDEVICE[VIDEO COMPOSING DEVICE] PROCESSINGUNIT <--> IMAGERETTOUCHINGDEVICE[IMAGE RETOUCHING DEVICE] PROCESSINGUNIT <--> GRAPHICSDRAWINGDEVICE[GRAPHICS DRAWING DEVICE] PROCESSINGUNIT <--> VIDEODISPLAYAPPARATUS[VIDEO DISPLAY APPARATUS] PROCESSINGUNIT <--> DYNAMICOBJECTBORDERTRACKER[DYNAMIC OBJECT BORDER TRACKER] DYNAMICOBJECTBORDERTRACKER <--> EFFECTDEVICE[EFFECT DEVICE] DYNAMICOBJECTBORDERTRACKER <--> IMAGERETTOUCHINGDEVICE </pre>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LJ	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

APPARATUS AND METHOD FOR OBJECT TRACKING

FIELD OF THE INVENTION

The present invention relates to image processing systems in general, and to object identification and tracking systems in particular.

BACKGROUND OF THE INVENTION

U.S. Patent 5,333,213 to Koyama et al. describes a method and apparatus for image region extraction, extracting an image of a moving object in a dynamic image.

U.S. Patent 5,067,014 to Bergen et al. describes a technique for analyzing two motions in successive image frames.

U.S. Patent 5,134,472 to Abe describes a method and apparatus for moving object detection.

Published European patent application EP 0 532 823 A2 describes a method for separating images.

U.S. Patent 5,274,453 to Maeda describes an image processing system using mask information to combine a plurality of images.

U.S. Patent 5,345,313 to Blank describes an image editing system which takes a background and inserts part of an image in the background.

U.S. Patent 5,093,869 to Alves et al. describes pattern recognition apparatus including high level graph matching.

Mathematical methods useful for image processing are described in the following references:

D. K. Ballard and C. M. Brown, Computer Vision, Prentice - Hall, 1982;

C. de Boor, A Practical Guide to Splines, New York, Springer-Verlag, 1978;

P. J. Schneider, "An algorithm for automatically fitting digitized curves", Graphic GEMs I, Academic Press, Inc.;

S. T. Barnard and W. B. Thompson, "Disparity analysis of images", IEEE transactions on pattern analysis and machine intelligence", PAMI - 2, No. 4, July 1980;

Yu-Ichi Ohta, Takeo Kanada, and T. Sakai, "Color Information for Region Segmentation", Computer Graphics and Image Processing 13, 222 241, 1980;

Yija Lin, Jiqing Dou and Eryi Zhang, "Edge expression based on tree structure", Pattern Recognition Vol. 25, No. 5, pp 507 - 517, 1992;

G. G. Pieroni and M. F. Costabile, "A method for detecting correspondences in a sequence of modifying shapes", Pattern Recognition Letters 3 (1985); and

R. N. Strickland and Zuhua Mao, "Computing Correspondences in a sequence of non-rigid shapes", Pattern Recognition, Vol. 25, No 9, 1992, pp. 901 - 912.

The disclosures of all of the above publications and all references cited therein are hereby incorporated herein by reference. The disclosure of all publications mentioned in the specification and all references cited therein are also hereby incorporated herein by reference.

SUMMARY OF THE INVENTION

The present invention seeks to provide an improved object identification and tracking system.

There is thus provided in accordance with a preferred embodiment of the present invention a tracking method including receiving a representation of an event including at least one dynamic object having a border and having at least one edge portion which is absent during at least a portion of the event, and providing an ongoing indication of the location of the border of the object during the event.

Further in accordance with a preferred embodiment of the present invention the representation includes a video representation.

Still further in accordance with a preferred embodiment of the present invention the edge portion includes a portion of the border.

Additionally in accordance with a preferred embodiment of the present invention the method also includes reconstructing at least one absent edge portion.

There is also provided in accordance with another preferred embodiment of the present invention a tracking method including receiving a representation of an event including at least one dynamic object having a border, and providing an ongoing indication of the location of the border of the object during the event.

There is also provided in accordance with another preferred embodiment of the present invention an edge-tracking method for tracking at least one dynamic object appearing in a sequence of frames, the method including for at least one key frame within the sequence of frames, marking at least one edge of at least one dynamic object based at least partly on external input, and for all frames within the sequence of frames other than the at least one key frame, automatically marking at least

one edge of at least one dynamic object based on output from the first marking step.

Further in accordance with a preferred embodiment of the present invention the method also includes remarking said at least one automatically marked edge at least once, based on external input.

Still further in accordance with a preferred embodiment of the present invention the external input includes human operator input.

Additionally in accordance with a preferred embodiment of the present invention at least one edge is marked without detecting the edge.

Moreover in accordance with a preferred embodiment of the present invention the at least one key frame includes a subsequence of frames preceding all other frames within the sequence.

Further in accordance with a preferred embodiment of the present invention the at least one key frame includes a subsequence of frames following all other frames within the sequence.

Still further in accordance with a preferred embodiment of the present invention the at least one key frame includes a subsequence of frames preceding at least one other frame within the sequence and following at least one other frame within the sequence.

There is also provided in accordance with another preferred embodiment of the present invention an edge-structuring method for structuring a plurality of connected edges into a graph, the method including providing a plurality of connected edges, traversing the plurality of connected edges in a chosen direction, and structuring the plurality of connected edges into a graph including a branch list and a node list, wherein the node list is independent of the chosen direction.

Further in accordance with a preferred embodiment of the present invention the node list includes an edge junction list.

Still further in accordance with a preferred embodiment of the present invention the node list includes an edge terminal point list.

Additionally in accordance with a preferred embodiment of the present invention the node list includes an edge corner list.

Moreover in accordance with a preferred embodiment of the present invention the node list includes a curvature list.

Further in accordance with a preferred embodiment of the present invention the plurality of connected edges includes a plurality of pixels and wherein the traversing step includes specifying a current pixel, identifying at least one visible pixel associated with the current pixel, and classifying the current pixel based, at least in part, on the number of visible pixels identified.

Still further in accordance with a preferred embodiment of the present invention the identifying step includes defining a blind strip, and ruling out as visible pixels at least one pixel associated with the blind strip.

Additionally in accordance with a preferred embodiment of the present invention the ruling out step includes ruling out as visible pixels all pixels associated with the blind strip whenever there is at least one visible pixel not associated with the blind strip.

There is also provided in accordance with another preferred embodiment of the present invention a method for tracking a border of a moving object, the method including selecting a plurality of border locations to be tracked in a first image, tracking at least some of the plurality of border locations from the first image to a second image, and computing the border in the second image based on an output of the tracking step and based on information characterizing the border in the first image.

Further in accordance with a preferred embodiment of the present invention at least one of the plurality of border

locations includes a location at which at least one border characteristic changes.

Still further in accordance with a preferred embodiment of the present invention the border characteristic includes at least one color adjacent to the border.

Additionally in accordance with a preferred embodiment of the present invention the tracking includes disregarding a border location which, when tracked from the first image to the second image, is found to have moved differently from other adjacent border locations.

Moreover in accordance with a preferred embodiment of the present invention the computing step includes transforming the border in the first image such that each of the plurality of border locations in the first image is transformed onto a corresponding one of the plurality of border locations in the second image.

Further in accordance with a preferred embodiment of the present invention the method also includes identifying an actual border in the second image by searching adjacent to the border as computed in the second image.

Still further in accordance with a preferred embodiment of the present invention an actual border is identified depending on whether the adjacent colors of the actual border resemble the adjacent colors of the border in the first image.

Additionally in accordance with a preferred embodiment of the present invention an output border is defined as the actual border, if identified, and as the border as computed in the second image, if no actual border is identified.

Moreover in accordance with a preferred embodiment of the present invention a first output border is defined which coincides in part with the actual border, where the actual border has been identified, and in part with the border as computed in the second image, where the actual border has not been identified.

Further in accordance with a preferred embodiment of the present invention the method also includes identifying a new

actual border in the second image by searching adjacent to the first output border, and defining a new output border which coincides in part with the new actual border, where the new actual border has been identified, and in part with the first output border, where the new actual border has not been identified.

Still further in accordance with a preferred embodiment of the present invention the transforming step includes transforming a spline representation of the border in the first image such that each of the plurality of border locations in the first image is transformed onto a corresponding one of the plurality of border locations in the second image.

Additionally in accordance with a preferred embodiment of the present invention the method also includes providing a first image seen from a first field of view and providing a second image seen from a different field of view.

Moreover in accordance with a preferred embodiment of the present invention the method also includes providing first and second images each including at least one of a moving dynamic object and a dynamic background.

Still further in accordance with a preferred embodiment of the present invention the automatic marking step includes automatically marking all edges of at least one dynamic object based on output from the first marking step.

There is also provided in accordance with another preferred embodiment of the present invention tracking apparatus including event input apparatus operative to receive a representation of an event including at least one dynamic object having a border and having at least one edge portion which is absent during at least a portion of the event, and a border locator operative to provide an ongoing indication of the location of the border of the object during the event.

There is also provided in accordance with another preferred embodiment of the present invention edge-tracking apparatus for tracking at least one dynamic object appearing in a sequence of frames, the apparatus including an edge marker

operative, for at least one key frame within the sequence of frames, to mark at least one edge of at least one dynamic object based at least partly on external input, and an automatic edge marker operative, for all frames within the sequence of frames other than the at least one key frame, to automatically mark at least one edge of at least one dynamic object based on output from the first marking step.

There is also provided in accordance with another preferred embodiment of the present invention edge-structuring apparatus for structuring a plurality of connected edges into a graph, the apparatus including an edge traverser operative to traverse the plurality of connected edges in a chosen direction, and a graph structurer operative to structure the plurality of connected edges into a graph including a branch list and a node list, wherein the node list is independent of the chosen direction.

There is also provided in accordance with another preferred embodiment of the present invention apparatus for tracking a border of a moving object, the apparatus including a border selector operative to select a plurality of border locations to be tracked in a first image, a border tracker operative to track at least some of the plurality of border locations from the first image to a second image, and border computation apparatus operative to compute the border in the second image based on an output of the border tracker and based on information characterizing the border in the first image.

There is also provided in accordance with another preferred embodiment of the present invention tracking apparatus including event input apparatus operative to receive a representation of an event including at least one dynamic object having a border, and a border locator operative to provide an ongoing indication of the location of the border of the object during the event.

Further in accordance with a preferred embodiment of the present invention, the method also includes generating an effect which is applied differentially on different sides of the

border.

Still further in accordance with a preferred embodiment of the present invention, the method also includes generating an effect which is applied differentially on different sides of the at least one edge.

Additionally in accordance with a preferred embodiment of the present invention, the effect includes an effect which is carried out at a location determined by a portion of the dynamic object.

Also provided, in accordance with another preferred embodiment of the present invention, is an image modification method including receiving a representation of an event, the representation including a plurality of frames, the event including at least one dynamic object having a border, computing the location of the border of the dynamic object during the event, generating an effect which is applied differentially on different sides of the border, and displaying a result of applying the effect without previously displaying a separate representation of the border.

Further in accordance with a preferred embodiment of the present invention, the step of generating an effect is performed on a subsequence of frames, including a plurality of frames, within the sequence of frames after an automatic marking step has been performed for the subsequence of frames.

Still further in accordance with a preferred embodiment of the present invention, the step of generating an effect is performed on an individual frame from among the sequence of frames after an automatic marking step has been performed for the individual frame.

Additionally in accordance with a preferred embodiment of the present invention, the effect is generated and displayed for an individual frame before the effect is generated for a subsequent frame.

Further in accordance with a preferred embodiment of the present invention, the effect is displayed for all of the plurality of individual frames without expecting user input

between frames.

Further in accordance with a preferred embodiment of the present invention, there is provided an image marking method including receiving a representation of an event, the representation including a plurality of frames, the event including at least one dynamic object having a border, computing the location of the border of the dynamic object during the event, and providing a user-sensible indication of locations of the dynamic object during the event, without previously displaying a separate representation of the border.

Still further in accordance with a preferred embodiment of the present invention, the effect includes one of the following group of effects: compositing, retouching, smoothing, compression, compositing, painting, blurring, sharpening, a filter operation, and an effect which changes over time at a different rate on different sides of the edge.

Additionally in accordance with a preferred embodiment of the present invention, the event includes a plurality of dynamic hotspot objects and wherein the providing step includes providing an ongoing indication of locations of borders of each of the plurality of dynamic hotspot objects during the event.

Further in accordance with a preferred embodiment of the present invention, the method also includes the steps of using the ongoing indication of locations of the borders of each of the hotspot objects to interpret a user's selection of an individual one of the plurality of dynamic hotspot objects, and displaying information regarding the individual dynamic hotspot object selected by the user.

Still further in accordance with a preferred embodiment of the present invention, the dynamic object is a portion of a larger object.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Fig. 1 is a simplified top-level block diagram of a dynamic object processing system constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 2A is a simplified flowchart of an interactive process for identifying boundaries of an object of interest in at least one key frame from among a sequence of frames and for marking the boundaries in the remaining frames from among the sequence of frames;

Figs. 2B - 2F are simplified pictorial illustrations of an example of rough marking in accordance with the method of steps 115, 130, 140, 150 of Fig. 2A;

Fig. 2G is a simplified flowchart of the process of Fig. 1 wherein the at least one key frame comprises only the first frame in the sequence of frames;

Fig. 3A is a simplified block diagram of apparatus, such as the dynamic object border tracker 70 of Fig. 1, for performing the method of Fig. 2A;

Figs. 3B - 3D are simplified pictorial illustrations showing internal junctions, external junctions, and occlusion;

Figs. 3E and 3F are simplified pictorial illustrations depicting a portion of the operation of step 370 of Fig. 3A;

Fig. 4 is a simplified block diagram of apparatus, such as the dynamic object border tracker 70 of Fig. 1, for performing the process of Fig. 2G wherein at least one key frame comprises the first frame in the sequence of frames;

Fig. 5 is a simplified block diagram of a modification of the apparatus of Fig. 3A in which borders are accurately identified;

Fig. 6 is a simplified block diagram of a modification of the apparatus of Fig. 4 in which borders are accurately

identified;

Fig. 7 is a simplified block diagram of a modification of the apparatus of Fig. 5 which is operative to predict border locations in non-key frames;

Fig. 8 is a simplified block diagram of a modification of the apparatus of Fig. 6 which is operative to predict border locations in non-key frames;

Fig. 9 is a simplified block diagram of a first alternative subsystem for performing the preprocessing operations of Figs. 3A and 4 - 8;

Fig. 10 is a simplified block diagram of the component mapping unit of Figs. 3A and 4 - 8;

Fig. 11A is a simplified block diagram of a preferred method of operation of units 1550 and 1560 of Fig. 10;

Figs. 11B and 11C are simplified pictorial illustrations of visible areas, useful in understanding the method of Fig. 11A;

Figs. 11D - 11H, are simplified pictorial illustrations of a plurality of pixels, useful in understanding the method of Fig. 11A;

Fig. 11I is a simplified pictorial illustration of an edge picture, from which a tree is to be built according to the method of Fig. 11A;

Fig. 12 is a simplified block diagram of the special points correspondence finding block of Figs. 3A and 4 - 8;

Fig. 13 is a simplified flowchart of a preferred method of operation for the special points weights computation unit 1700 of Fig. 12;

Fig. 14 is a simplified flowchart of a preferred method of operation for the border estimation block of Figs. 3A and 4;

Fig. 15 is a simplified flowchart of an alternative preferred method of operation for the border estimation block of Figs. 3A and 4;

Fig. 16 is a simplified flowchart of a preferred method of operation for the borders and mask generation unit of Figs. 3A and 4 - 8;

Fig. 17 is a simplified flowchart of a preferred method of operation for the exact object border description blocks of Figs. 3A, 4, 5, 6, 7 and 8;

Fig. 18 is a simplified flowchart of a preferred method of operation of steps 570, 572, and 574 of Fig. 5 and of steps 670, 672, and 674 of Fig. 6;

Fig. 19 is a simplified flowchart of an alternative method of operation of step 2340 of Fig. 18;

Fig. 20 is a simplified flowchart of a prediction method useful in the methods of Figs. 7 and 8;

Fig. 21 is a simplified flowchart of a preferred method for carrying out the steps of Fig. 20 in the case of first-order prediction;

Fig. 22 is a simplified flowchart of a preferred method for carrying out the steps of Fig. 20 in the case of second-order prediction; and

Fig. 23 is a simplified flowchart of a preferred method for carrying out the steps of Fig. 20 in the case of third-order and higher prediction.

Fig. 24 is a simplified block diagram of a modification of the apparatus of Fig. 4;

Fig. 25 is a simplified block diagram of a modification of the apparatus of Fig. 8;

Fig. 26 is a simplified block diagram of a modification of the apparatus of Fig. 3A; and

Fig. 27 is a simplified block diagram of a modification of the apparatus of Fig. 7.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Reference is now made to Fig. 1 which is a simplified top-level block diagram of a dynamic object processing system constructed and operative in accordance with a preferred embodiment of the present invention. The term "dynamic object" is here intended to include objects which are stationary at times and at motion at other times, as well as objects which are always in motion.

The system of Fig. 1 receives a sequence of time-varying images such as animation, photographed or other video images from any suitable source, via a suitable video interface 10 which may include an A/D unit if the input thereto is analog. Suitable video sources include, for example, a video camera 20, a network, a video storage unit 30 (video memory, video disk, tape, CD-ROM or hard disk) or a film scanner 40.

The system includes processing unit 50, associated with a video memory 54. The processing unit 50 may, for example, be any appropriate computer equipped with video capability and programmed with appropriate software. For example, an IBM compatible Pentium PC, equipped with video I/O cards, as are well known in the art, may be used. Alternatively, the processing unit 50 may be implemented partly or completely in custom hardware or otherwise.

The processing unit 50 receives from a suitable user input device such as a graphics drawing device 60 (e.g. tablet and stylus or mouse), an indication of at least one initial border of at least one dynamic object, in an initial frame. Alternatively, the indication may be of borders of the dynamic object as it appears other than in the initial frame.

The term "frame", as used throughout the present specification and claims, refers to either a frame as generally understood in the art or, in the case of interlaced video wherein a frame as generally understood in the art comprises more than one field, any of the fields comprising a frame as generally

understood in the art.

The frame or frames for which the user provides a border indication (here termed "a reference border") are termed herein "key frames". Preferably, key frames are selected to be those frames in which a characteristic of the dynamic object's appearance changes, e.g. due to a change in the object's motion or due to occlusion by another object or due to light condition changes.

It is appreciated that the frames, whether key frames or non-key frames, may comprise a plurality of frames seen from more than one field of view, such as, for example, two different fields of view, or a dynamic field of view.

It is appreciated that the frames may comprise frames depicting a dynamic object, a dynamic background, or both.

The processing unit 50 includes a dynamic object border tracker 70 which is operative to track the borders of the dynamic object through non-key frames, based on the locations of the borders in the key frames. It is appreciated that the dynamic object border tracker 70 may preferably be operative to track borders in any direction through the non-key frames, that is, forwards, backwards, converging from both ends, and so forth.

Preferably, the dynamic object border tracker 70 is operative to complete a border by adding border segments which the tracker 70 did not succeed in finding. These border segments are termed herein "missing border segments".

The user may interactively correct the tracking of the border through either key frames or non-key frames by means of the drawing device 60.

The output of the dynamic object border tracker 70 typically comprises an indication of the location of the border for each of the frames of the image sequence. The border location indication typically comprises a mask, having "1" values at the border and "0" values other than at the border. The border location indication is fed to and utilized by any of a plurality of application devices, thereby enabling an operator to issue a single command for processing the dynamic object in the entire

image sequence, rather than having to process the dynamic object "frame by frame", i.e. separately for each frame. Similarly, processing of the background in the entire image sequence may also be carried out without having to process separately for each frame.

Examples of suitable application devices include:

- a. A video compositing device 80, operative to generate a video image comprising a plurality of "layers".
- b. An image retouching device 90, operative to perform one-step enhancement, segmentation or special effects, on at least one dynamic object in the image sequence, rather than frame-by-frame retouching of the dynamic object. Retouching operations include: color alteration, filtering, as, for example, noise reduction, sharpening, or other types of filtering; and effects, as, for example, tiling.

Alternatively, the border location may be fed elsewhere, as, for example, to the network or to the video storage unit 30.

Preferably, a video display device 95 provides a display which facilitates interactive sessions.

It is believed that the border location indication may also be employed for a variety of other applications, including, for example, the following:

- a. video rate conversion or video standard conversion.
- b. image compression in which at least one dynamic object in the image is compressed differently, typically more accurately, than the remaining portions of the image.
- c. Scene analysis, such as automatic navigation applications in which the borders of encountered objects are tracked so as to determine an optimal route therepast.

Reference is now made to Fig. 2A which is a simplified flowchart for interactive operation of the dynamic object border tracker of Fig. 1. In the method of Fig. 2A, boundaries of an object of interest are identified in at least one key frame from among a sequence of frames and are utilized for marking the boundaries in the remaining frames from among the sequence of

frames.

The user may select or localize borders by any suitable method such as:

a. As shown in Fig. 2A (step 115), rough manual marking of border location of an object of interest, e.g. by means of a brush operated by a tablet's stylus, such as a stylus associated with the graphics drawing device 60. The system then attempts to find a plurality of candidate edges within the rough marking. These candidate edges are displayed to the user who selects an appropriate edge from among them.

Alternatively, the following border selection or localization methods may, for example, be employed:

b. The user may mark the exact border location manually.

c. A spline tool or other curve drawing tool may be employed by the user to mark the border location.

d. The user may select a border contour from among a library of border contours such as rectangles, previously used border contours or other predetermined border contours.

e. The user may use another means of indicating the border as, for example, choosing a color selecting method well known in the art such as chroma-key or color-key, to identify either the object or the background. The system then identifies the transition between the selected and unselected portions, using methods well-known in the art, and takes the transition between the selected and unselected portions to be the border.

f. Any combination of (a) - (e).

In the case of (b) - (d), the system may, preferably at user option, add a rough marking surrounding all or a portion of the marking selected by the user. The result of this process is a rough marking similar to that of case (a), and the rough marking is then utilized as described above for case (a).

Missing edges may preferably be filled in by the user.

The borders, once marked as above, may preferably be modified manually by the user.

The system finds the marked border locations in key frames (step 130), and gives the user the option to modify the

marked border locations based on the system response (steps 140 and 150).

Reference is now additionally made to Figs. 2B - 2F which are simplified pictorial illustrations of an example of rough marking in accordance with the method of steps 115, 130, 140, and 150 of Fig. 2A. In Figs. 2B - 2F, option (a), rough manual marking, is shown. Figs. 2B - 2F depict a display preferably provided to the user during the operation of steps 115, 130, 140, and 150 typically on video display 95 of Fig. 1.

Fig. 2B depicts an actual frame. Preferably, in Figs. 2C - 2F the actual frame of Fig. 2B is displayed as background to assist the user in making marks and modifications. For simplicity, said background is not shown in Figs. 2D - 2F.

Fig. 2B comprises a plurality of edges 116. The edges 116 comprise the limits of areas 121, 122, 123, 124, and 125. In Fig. 2B, the areas 121, 122, 123, 124, and 125 are taken to be of different color, but it is appreciated that, in general, different areas need not be of different color. Areas 125 are taken not to be of interest to the user, while areas 121, 122, 123 are areas of interest, because areas of interest 121, 123, and 123 together comprise a desired object 117. Area 124 is also taken to be not of interest to the user. In the example of Figs. 2B - 2F, areas are defined by being surrounded by closed edges, or by the ends of the video display 95.

In Fig. 2C, the user has marked a rough marking area 126, whose limits are shown in Fig. 2C by marking area limits 127. Typically, the user marks the rough marking area 126 with, for example, the graphic drawing device 60. Alternatively, the user may mark the marking area limits 127 and indicate that the area 126 in between the marking area limits 127 is to be the rough marking area.

In Fig. 2D, all edges 116 not within the rough marking area 126 have been removed by step 130 of Fig. 2A. It is appreciated that the rough marking area 126 includes areas of interest 121, 122, and 123 and the edges 116 surrounding them, along with area 124, which is not of interest.

In Fig. 2E only the remaining edges, within the rough marking area 126 are shown. Edge 128 is internal to the desired object 117, while edges 129 and 130 define area 124, which is not of interest. In order to define the areas of interest based on closed edges, it is desirable to open the edges of area 124, so that area 124 will be outside the desired object 117. Typically, the user decision of step 140 is based on the display of Fig. 2E.

In Fig. 2F, representing the results of step 150, the user has erased a portion of edge 129, typically using the graphic drawing device 60, so that area 124 is now outside the desired object 117. It is appreciated that, in general, the user may make a wide variety of modifications, including erasing edges or portions thereof, and adding edges or portions thereof.

In step 155, the system learns the border qualities of the key frames. Border qualities may comprise, for example, border length, average color and color changes. The border qualities may also comprise aspects of motion of the borders such as, for example, border velocity and border acceleration. The method of step 155 is explained in more detail below with reference to the apparatus of Fig. 3A.

Non-key frames are input in step 190.

The system then proceeds to identify, in non-key frames, the borders which were marked in the key frames (step 160). In this step, the system may optionally make use of information obtained from the processing of other frames which were already processed. Preferably, the system seeks the borders only in a specific region of interest (ROI), which is taken to be the region in which the borders are expected to be found, typically based on information from other frames already processed, by identifying a region around the object borders in the previously processed frame or frames as the ROI. The method of step 160 is described in more detail below with reference to the apparatus of Fig. 3A.

It is appreciated that the borders output in step 160 may be fed back to step 155, where the borders may be treated as

key frame borders in further iterations of the method of Fig. 2A.

If the user decides that the object borders in the non-key frames found by the system in step 160 are not good enough (step 170), the user may modify these borders directly (step 180) or can decide to define more of the frames, or different frames, as key-frames and re-run the system, based on the reference borders of the new set of key-frames. Typically, in step 180, the user is presented with displays and modification options similar to those described above with reference to Figs. 2E and 2F. Alternatively, the user may use any other method as described with reference to step 115 of Fig. 2A.

The user may be confronted with the "good enough?" decision of step 170 either only once, for the image sequence as a whole, or at one or more intermediate points, determined by the user and/or by the system, within the process of border marking of the image sequence.

The user may make a "good enough?" decision regarding each of the frames of the image sequence or regarding only some of the frames of the image sequence.

It is appreciated that, throughout the operation of the method of Fig. 2A, the user may provide additional input, to modify or correct the operation of the system, at any of the steps involving user input, comprising steps 110, 150, and 180.

Fig. 2G is a special case of the flowchart of Fig. 2A in which a first frame in a frame sequence is initially selected as the sole key frame, and processing continues on all the frames. The steps of Fig. 2G are self-explanatory in light of the above explanation of Fig. 2A, except as follows.

In step 210, the system may input only one frame or a sequence of sequential frames. In the case where a sequence of frames is input, steps 220, 230, 240, and 250 process the sequence of frames, typically one frame at a time.

In step 255 the system learns the border qualities of the current frame. Optionally, border qualities of a sequence of frames may be learned as, for example, border length, average color and color changes. Such a sequence may have been input in

step 210 or may be built in step 255 as a plurality of frames is processed. In the case where a sequence of frames is processed, the border qualities may also comprise aspects of motion of the borders such as, for example, border velocity, border acceleration. The method of step 255 is explained in more detail below with reference to the apparatus of Fig. 4.

In step 290, the next frame is input. The system finds borders of the next frame in step 260 with reference to the border qualities learned in step 255. As described above with reference to step 160, the operation of step 260 is preferably limited to an ROI, defined based on the object behavior in previous frames. The operation of step 260 is further described below with reference to the apparatus of Fig. 4.

After step 275, if the last frame has not been reached, processing continues with step 255.

Reference is now made to Fig. 3A, which is a simplified block diagram of apparatus, such as the dynamic object border tracker 70 of Fig. 1, for performing the method of Fig. 2A.

The steps of Fig. 2A are performed by the following units in Fig. 3A:

Step 110	Unit 310
Step 115	Unit 320
Step 130	Units 330 and 340
Step 140	Unit 335
Step 150	Unit 320
Step 190	Unit 355
Step 155	Units 350 and 380
Step 160	Units 330, 340, 360 and 370
Step 170	Unit 379
Step 180	Unit 320

In a pre-processing unit 330, edges in each frame, including key frames and non-key frames, are detected and are preferably modified to facilitate the remaining steps, as described in more detail below with reference to Fig. 9.

In a component mapping unit 340, the edges found by the pre-processing unit 330 are traced, as further described below

with reference to Fig. 10, and a data structure is generated to represent the edges. This structure typically comprises a forest of edge trees in which each branch comprises an edge and each node comprises a "special point" such as, for example, a junction. Special points may also, for example, include terminal points in an edge, whether or not the edge is connected to a junction, and edge corners.

As used throughout the present specification and claims, the term "tree" includes trees which contain loops, that is, paths which return to a junction that was already reached. It is appreciated, as is well known in the art, that this type of tree may also be depicted as a graph. Thus all operations specified herein to be performed on a tree may be performed on a corresponding graph. The terms "graph" and "tree" as used throughout the specification and claims are each meant to include both graph and tree representations.

The term "forest" is used throughout the present specification and claims to refer to a collection of trees. It is appreciated that a single graph may represent a collection of trees, and thus a forest may comprise a single graph or more than one graph.

Preferably, when tracking an object, special points are taken to be internal junctions, that is, junctions internal to the object or lying on the internal side of the border thereof. In the case where one object occludes another, tracking external junctions may be preferred; alternatively, a combination of internal and external junctions or other special points may be tracked, depending on the precise position of a plurality of partially occluding objects.

Reference is now additionally made to Figs. 3B - 3D which are simplified pictorial illustrations showing internal junctions, external junctions, and occlusion. Fig. 3B comprises a tracked object 341 and a second object 342. The tracked object 341 has internal junctions 343 and 344 and an external junction 345. The second object 342 has an internal junction 346. It is appreciated that there may be other junctions in addition to

those which are shown in Fig. 3B. As stated above, special points are, preferable, taken to be internal junctions.

In Fig. 3C the tracked object 341 and the second object 342 have moved closer to each other, such that the second object 342 partially occludes the tracked object 341. It is appreciated from Fig. 3C that internal junction 344, visible in Fig. 3B, is not visible in Fig. 3C due to the partial occlusion of the tracked object 341. In addition, new external junctions 347 and 348 are created due to the partial occlusion. Furthermore, it will be seen that, in Fig. 3C, junction 346 is now an external junction of the tracked object 341 due to the partial occlusion. It will therefore be appreciated that, in the case of partial occlusion, it may be preferred to take external junctions also as special points.

In Fig. 3D the tracked object 341 and the second object 342 have moved still closer to one another, such that the extent of occlusion is greater. It is appreciated that, in Fig. 3D, new external junctions 347 and 348 and external junction 346 are still present, so that designating junctions 346, 347, and 348 as special points would be preferred in tracking the tracked object 341.

Unit 340 is described in more detail below with reference to Figs. 10 and 11.

Unit 350 receives the output of step 340, and creates an exact object border description, preferably represented in terms of a "chain code", for each keyframe. A chain code is a representation of the border in terms of edges and special points and typically comprises pointers to the edges and special points which form the border, in their proper sequence.

In addition curves, typically splines, connecting the points, are computed. These curves are also termed herein "initial border estimation segments". Computation of splines is described in C. de Boor and in P. J. Schneider, referred to above. The splines are typically employed in further steps for the purpose of border estimation.

The chain code and a representation of the splines,

typically a representation of the control points of the splines' control polygons, are stored in the data base 380.

The operation of unit 350 is described more fully below, with reference to Fig. 17.

Typically after the keyframes have all been processed, units 330 and 340 operate on the non-keyframes.

Unit 360 finds correspondences between the special points in the keyframe or frames and special points in the non-keyframes, that is, pairs of special points which, one in a key frame and one not in a key frame, represent the same location in the object. In the context of processing other frames, corresponding special points may be treated as estimated special points.

The correspondence is found with reference to stored point data found in the data base 380. The stored point data typically comprises chain codes representing special points and edge segments of borders and spline representations of edge segments of borders, both produced by units 350 and 370. Optionally, once correspondences are found between special points in a keyframe and special points in a non-keyframe, these correspondences may be employed to find correspondences between special points in the keyframe and special points in other non-keyframes. Unit 360 is described in more detail below with reference to Figs. 12 and 13.

When a correspondence is found with reference to, for example, two special points, the special points are termed herein "corresponding points.". Similarly, when a correspondence is found between two border segments, the two border segments are termed herein "corresponding segments". Corresponding points and corresponding segments are assumed to represent the same points and border segments, respectively, in the dynamic object. The process of finding corresponding segments is described below with reference to unit 370.

Preferably, the operation of unit 360 is restricted to an ROI, as described above with reference to Fig. 2A. In step 360, the ROI is typically taken to be a region of a predetermined

size around the borders and special points of the object as, for example, five pixels around.

The special points identified by unit 360 are received by unit 370, and the chain codes and spline representations stored in the data base 380 are retrieved. When a gap exists because certain special points were not found by unit 360, gaps are filled in via use of the spline representations, which represent border estimation.

The points are typically connected together by projecting the spline curve, from the data base 380, of the initial border estimation segments between the points. Typically, since the points may have moved and the distance between the points and their position relative to other points may have changed since the spline curve was stored in the data base 380, the projecting preferably includes use of an affine transformation. The affine transformation may include rotation, scaling, and shifting. Affine transformations are well known in the art, and are described in Ballard and Brown, referred to above, at page 477.

Preferably, the affine transformation is applied only to the control points of the spline curve, and the spline curve is then recomputed.

The newly projected curves, when applied to other frames, are termed herein "estimated border segments".

Reference is now additionally made to Figs. 3E and 3F which are simplified pictorial illustrations depicting a portion of the operation of step 370 of Fig. 3A.

Fig. 3E depicts a first frame, comprising special points 371, 372, 373 and 374. Fig. 3F depicts another frame, comprising special points 375, 376, and 377. Correspondences have already been found, as described above with reference to unit 360, between the following pairs of special points: 371 and 375; 372 and 376; and 373 and 377. No correspondence was found for point 374.

Estimated border segments 378, projected as previously described, has been added between each adjacent pair of points,

including points 375 and 377. It is appreciated that an estimated border segment 378 is projected between points 375 and 377 even though no corresponding point was found for point 374, based on the previous segments between points 373, 374, and 371.

Updated chain codes are computed from the estimated border segments and the corresponding special points. Descriptions of the special points and estimated border segments, as well as the updated chain codes, are stored in the data base 380. Estimated border segments may be used, in a later iteration, as initial border estimation segments. Computation of chain codes is described in more detail below with reference to Fig. 17.

An object border description, comprising an externally-useable representation of the object border as, for example, a list of coordinates defining the location of the border, and an object mask, suitable for further processing, are generated by unit 390. Alternatively, only an object mask may be generated, and an object border description may be generated from the object mask in a later step when the object border description is to be used. Unit 390 is more fully described below with reference to Fig. 16.

Unit 379 allows the user to examine the results of the method of Fig. 3A and to modify the results accordingly, including choosing new key frames, as explained above with reference to Figs. 2A - 2F. The results of the method of Fig. 3A are presented to the user by drawing the object chain code over the present frame. It is appreciated that, although unit 379 is depicted as receiving input from unit 370, unit 379 may also utilize any other object border information available as, for example, information stored in the data base 380.

Unit 335 operates similarly to unit 379, except that unit 335 relates to preprocessing and thus it draws directly the edges found by unit 330 rather than using a chain code drawing.

Reference is now made to Fig. 4 which is a simplified block diagram of apparatus for performing the process of Fig. 2G

wherein at least one key frame comprises the first frame in the sequence of frames.

The steps of Fig. 2G are performed by the following units in Fig. 4:

Step 210	Unit 410
Step 220	Unit 420
Step 230	Units 430 and 440
Step 240	Unit 435
Step 250	Unit 420
Step 290	Unit 455
Step 255	Units 450, 480, 465
Step 260	Units 430, 440, 460 and 470
Step 270	Unit 478
Step 280	Unit 420

The units of Fig. 4 are similar to the units of Fig. 3A and are self-explanatory with reference to the above discussion of Fig. 3A, except as described below. The correspondence between the units of Fig. 3A and the units of Fig. 4 is as follows:

Unit 310	Unit 410
Unit 320	Unit 420
Unit 330	Unit 430
Unit 340	Unit 440
Unit 335	Unit 435
Unit 350	Unit 450
Unit 355	Unit 455
Unit 360	Unit 460
Unit 370	Unit 470 and Unit 465 combined
Unit 379	Unit 478
Unit 380	Unit 480
Unit 390	Unit 490

In Fig. 4, the following units operate on consecutive first frames, treating the first frames as key frames, rather than on key frames in general as in the corresponding units of Fig. 3: 410, 435, 450.

In Fig. 4 a plurality of first frames are processed

together.

Unit 455 provides next frames consecutively, preferably one frame at a time, to unit 430.

Preferably, the operation of unit 460 is restricted to an ROI, as described above with reference to Fig. 2G. In unit 460, the ROI is typically taken to be a region of an predetermined size around the borders and special points of the object as, for example, five pixels around. Preferably, unit 460 operates on consecutive frames, one frame at a time.

Unit 460 finds correspondences between the special points in consecutive frames, that is, pairs of special points which, one in a first frame and one in the succeeding frame, represent the same location in the object. In the context of processing further frames, corresponding special points may be treated as estimated special points.

The correspondence is found with reference to stored point data found in the data base 480. The stored point data typically comprises chain codes representing special points and edge segments of borders and spline representations of edge segments of borders, both produced by units 450 and 465, described below. Optionally, once correspondences are found between special points in two consecutive frames, these correspondences may be employed to find correspondences between special points in the two frames and special points in other frames. Unit 460 is described in more detail below with reference to Figs. 12 and 13.

Unit 470 operates similarly to unit 370, except that an exact object border description is created by unit 465. A current frame chain code, representing an exact border description of the current frame, is computed based on the corresponding special points found by unit 460 and the borders estimated by unit 470. It is appreciated that, in Fig. 3A, the functionality of unit 465 is included in unit 370. Unit 465 is described more fully below with reference to Fig. 17.

Unit 478 operates similarly to unit 379 of Fig. 3A, except that unit 478 operates on the output of unit 465.

Reference is now made to Fig. 5 which is a simplified block diagram of a modification of the apparatus of Fig. 3A in which borders are accurately identified. The units of Fig. 5 are self-explanatory with reference to the above discussion of Fig. 3A, except as follows.

In Fig. 5, additional processing is performed after the completion of border estimation. An estimated border, represented by a chain code termed herein an "intermediate chain code", is created by unit 570. A more precise border is identified by unit 572, based on the estimated border produced by unit 570 and on chain code and spline data describing a stored frame border, obtained from the data base 580. Unit 572 preferably operates by identifying edges in the vicinity of the estimated border and selecting the best candidates for border segments. Estimated border segments provided by unit 572 may be filled in by unit 574 where a more precise border was not successfully identified by unit 572, and an exact object border description is created.

Preferably, the operation of unit 572 is restricted to an ROI, as described above with reference to Fig. 2A. In unit 572, the ROI is typically taken to be a region of an predetermined size around the borders and special points of the object as, for example, five pixels around.

Units 570, 572 and 574 are described more fully below with reference to Figs. 18 and 24.

A chain code is computed based on the new more precise border by unit 576. In the case of Fig. 5, the chain code is typically computed by unit 576 rather than by unit 570. The chain code is stored in the database 580 and is also passed along to unit 578, which allows the user to examine the new border. The operation of unit 576 is described in more detail below with reference to Fig. 17.

Reference is now made to Fig. 6 which is a simplified block diagram of a modification of the apparatus of Fig. 4 in which borders are accurately identified. The units of Fig. 6

are self-explanatory with reference to the above discussion of Fig. 4, except as follows.

An estimated border, represented by an intermediate chain code, is identified by unit 670. A more precise border is identified by unit 672, based on the estimated border from unit 670 and on data on previous frames, preferably comprising chain codes and splines, obtained from the data base 680. Unit 672 preferably operates by identifying edges in the vicinity of the estimated border and selecting the best candidates for border segments. The operation of unit 672 is described in detail below with reference to Figs. 18 and 24.

Preferably, the operation of unit 672 is restricted to an ROI, as described above with reference to Fig. 2A. In unit 672, the ROI is typically taken to be a region of an predetermined size around the borders and special points of the object as, for example, five pixels around.

In unit 674, estimated border segments may be filled in where a more precise border was not successfully identified by unit 672, and an exact object border description is created by unit 665. The operation of unit 674 is described in detail below with reference to Figs. 18 and 24.

Reference is now made to Fig. 7 which is a simplified block diagram of a modification of the apparatus of Fig. 5 which is operative to predict border locations in non-key frames. The units of Fig. 7 are self-explanatory with reference to the above discussion of Fig. 5, except as follows.

Unit 774, exact object border description, performs both the operation of unit 574 of Fig. 5 and the operation of unit 576 of Fig. 5.

Unit 777 applies equations of motion, relating position to changes in position and to rate of change in position, to the positions of special points and borders stored in the data base 780 in order to predict the location, in upcoming frames, of the special points and borders. It is appreciated that, in applying the equations of motion, it is necessary to take into account the distance and direction in time, in frames, between key frames

being processed, since time between frames is an important variable in applying equations of motion. Equations of motion are discussed in more detail below with reference to Figs. 21 - 23.

Unit 752 operates similarly to unit 777, but uses equations of motion to predict special points and borders according to the key frames rather than using other frames as is the case with unit 752.

Similarly, unit 771, in contrast to unit 570 of Fig. 5, may apply equations of motion also to the stored spline data received from the data base 780, so that the stored spline data is updated to more accurately predict border position.

Reference is now made to Fig. 8 which is a simplified block diagram of a modification of the apparatus of Fig. 6 which is operative to predict border locations in non-key frames. The units of Fig. 8 are self-explanatory with reference to the above discussion of Figs. 6 and 7, except as follows.

A prediction is made of the special points and borders by unit 866. Equations of motion are applied to the positions of special points and borders for a previous frame stored in the data base 880 in order to predict the location, in subsequent frames, of the special points and borders. Equations of motion useful in the method of Fig. 8 are discussed below with reference to Figs. 20 - 23.

Unit 852, similarly to unit 866, uses equations of motion to predict special points and borders according to the key frames.

Similarly, unit 871, like unit 771 of Fig. 7, may apply equations of motion to the stored spline data received from the data base 880, so that the stored spline data is updated to more accurately predict border position.

Unit 874 is similar to unit 774 of Fig. 7.

Reference is now made to Fig. 9 which is a simplified block diagram of an alternative subsystem for performing the preprocessing operations of blocks 330, 430, 530, 630, 730, and 830 of Figs. 3A and 4 - 8.

A commonly used RGB color space may not be optimal for edge detection because the three components R-G-B tend to all change similarly and in concert with intensity change, so that edges identified from the components of such a color space will tend to be similar. It is therefore desirable to choose a color space where the above behavior typically does not occur, that is, where the components tend to behave differently, so that edges identified from the components of such a color space will tend to be different. Preferably, a color space having the following components, computed from R, G, and B components of an RGB color space, is used:

$$I_1 = (R + G + B) / 3$$

$$I_2 = (R - B)$$

$$I_3 = (2 * G - R - B) / 2$$

The above color space is discussed in Yu-Ichi Ohta, Takeo Kanada, and T. Sakai, referred to above.

In Fig. 9, the input color components are converted to the chosen color space (unit 1100). In the case of the color space described above, the formulas provided may be used to compute the conversion. Optionally, the RGB space or any other appropriate color space may be used.

Edge detection for each color component is then performed (unit 1110). In order to eliminate falsely detected edges, a minimum threshold value is applied to the color intensity of each color component and all edges whose color component intensity is less than the threshold value are ignored (1120).

Edges detected in the separate color components are merged together (unit 1130). An edge picture, comprising typically "1" values wherever an edge was detected and "0" values otherwise, is produced.

Reference is now made to Fig. 10 which is a simplified block diagram of the component mapping unit of Figs. 3A and 4 - 8. Unit 1500 makes a working copy of the edge picture produced by the pre-processing units of Figs. 3A and 4 - 8. The working copy is scanned for an edge pixel (unit 1510), until the end of

the picture is reached (unit 1520). If the current pixel is not an edge pixel (unit 1540), scanning continues.

If the current pixel is an edge pixel, pixels along the edge are traversed until a junction pixel, a terminal pixel, or another special point is identified as described below with reference to Fig. 11A, and the junction pixel, terminal pixel, or other special point is identified as a root pixel. All pixels connected with the root pixel are traced, forming an edge tree (unit 1550). If no junction pixel, terminal pixel, or other special point is found, the initial edge pixel is taken as the root pixel.

Unit 1550 identifies candidate special points, as, for example, points at edge junctions. Candidate special points may also, for example, include terminal points in an edge not connected to a junction and edge corners.

The edge tree is added to an edge forest consisting of all edge trees found (step 1570), and the pixels of the edge tree are erased from the working copy of the edge picture (step 1560).

The edge forest provides a component map comprising special point candidates and edge candidates and the relationships between them, as described below with reference to Fig. 11A.

Methods for forming an edge tree are well known in the art, and include the method described in Yija Lin, Jiqing Dou and Eryi Zhang, "Edge expression based on tree structure", Pattern Recognition Vol. 25, No. 5, pp 507 - 517, 1992, referred to above.

The methods for forming an edge tree known in the art have the drawback that the list of edges and the list of nodes produced are not necessarily independent of the direction of traversal of the edges and of the choice of root node. A preferred method for forming an edge tree, which overcomes the drawbacks of methods known in the prior art, is now described with reference to Fig. 11A as follows. The method of Fig. 11A is specific to the case where all special points are junctions.

Evaluation rules for forming the edge tree are as

follows:

Visible area rule: The region around the current edge pixel, as seen from the direction of entry to the pixel and towards the other directions, is termed herein a "visible area". The visible area of the current edge pixel is classified as diagonal or straight according to the direction in which the current edge proceeds from the current edge pixel. "Straight" means entering horizontally or vertically, that is, not diagonally. Reference is now additionally made to Figs. 11B and 11C, which are simplified pictorial illustrations of visible areas, useful in understanding the method of Fig. 11A. In Fig. 11B, arrows depict the directions that are straight, while in Fig. 11C arrows depict the directions which are diagonal.

All directions are seen as part of the visible area except as follows:

a. If the visible area is straight, all backwards directions, both the directly backwards direction and the diagonally backward directions, are not seen.

b. If the visible area is diagonal, only the directly backwards direction, which is the diagonal direction from which the pixel was entered, is not seen.

Reference is now additionally made to Figs. 11D - 11E, which are simplified pictorial illustrations of a plurality of pixels, useful in understanding the method of Fig. 11A. In Figs. 11D and 11E, arrows depict the direction of entry into the visible area. Fig. 11D comprises a straight visible area. Fig. 11E comprises a diagonal visible area.

Blind strip rule: If, in the visible area of the current edge pixel, there are one or more pixels in a straight direction, further connected edge pixels are preferably sought in the straight direction, and the diagonal directions are blocked, in the sense that they are not seen as part of the visible area.

Reference is now additionally made to Figs. 11F - 11H, which are simplified pictorial illustrations of a plurality of pixels, useful in understanding the method of Fig. 11A. Figs. 11F - 11H comprise a plurality of edge pixels and depict

application of the blind strip rule thereto. In Figs. 11F - 11H, arrows depict the directions in which additional pixels are sought according to the blind strip rule. Each of Figs. 11F - 11H depict entry at a different pixel. It is appreciated that, in each case, regardless of point of entry, the same junctions are found.

Visible pixel rule: The term "visible pixels", as used throughout the specification and claims, refers to edge pixels adjacent to the current pixel in the visible area, not including any pixels ignored under the blind strip rule. Note that, generally, because of the method of identifying edge pixels, no more than 3 visible pixels will be seen, except in the case of a root pixel which is at a junction of four edges, in which case 4 visible pixels will be seen.

The current edge pixel is classified based on the following pixel classification rules:

1. If the current pixel has two or more visible pixels, the current pixel is identified as a junction pixel. However, if exactly 2 visible pixels are seen and the current pixel is a root pixel, the current pixel is not identified as a junction pixel; rather, subsequent pixels are processed.

2. If the current pixel has no visible pixels, the current pixel is identified as a terminal pixel.

3. If the current pixel has one visible pixel, the current pixel is identified as a "usual branch pixel". However, if the current pixel is the root pixel, the current pixel is classified as a terminal pixel.

The description of the tree structure is as follows:

Every element of the tree is either a branch or a junction. A branch is defined herein as a sequential, connected set of usual branch pixels, and is typically represented as a dynamic array of pixel coordinates and characteristics. The color typically represents the color or some other characteristic of the pixel, such as an indication that the pixel is a pixel which was added to fill in a gap.

Each element of the tree is preferably defined by a

list of attributes, preferably including the following:
flag, defining its type as branch or junction;
parent pointer, pointing to previous element or
parent; and
neighboring pointers, pointing to neighboring
elements in the direction of traversal, or children.

The tree is then built according to the following tree
building method:

First, the first pixel is classified according to the
above rules (step 1630).

1. Delete parent pixel (step 1650); i.e., delete pixels
from the image that were already processed. This step is omitted
in case of the first pixel, which has no parent.

2. Exception to rule 1: if current pixel is a
junction (step 1640), perform a depth-first search and delete a
junction only after all of its children are evaluated (steps 1660
and 1670).

3. If at least one visible edge pixel exists (step
1610), move forward to the next edge pixel (1620).

4. Classify the next pixel (step 1630).

It is appreciated that, in addition to junctions, other
types of special points may be identified as, for example,
corners.

Reference is now additionally made to Fig. 11I, which
is a simplified pictorial illustration of an edge picture, from
which a tree is to be built according to the method of Fig. 11A.

Fig. 11I comprises an edge 1690. Fig 11I also
comprises an edge junction 1691, at the end of the edge 1690.
Fig. 11I also comprises an edge 1692 lying between the edge
junction 1691 and an edge junction 1693. Fig. 11I also comprises
an edge 1694, at one end of which lies edge junction 1693. Fig.
11I also comprises an edge 1695, lying between edge junction 1691
and edge junction 1693.

Processing of Fig. 11I in accordance with the method of
Fig. 11A, in order to build a tree, may proceed as follows:

1. Begin at a root pixel (not shown in Fig. 11I) at

the end of edge 1690 away from the edge junction 1691.

2. According to pixel classification rule 3, the root pixel is classified as a terminal pixel (step 1630).

3. All of the pixels of edge 1690 are processed according to steps 1610, 1620, 1630, 1640 and 1650 until the edge junction 1691 is reached.

4. An edge pixel (not shown in Fig. 11I) is found at edge junction 1691, and processing continues with steps 1660 and 1670. The effect of steps 1660 and 1670, which comprise depth-first search processing and junction deletion, is to process the remainder of Fig. 11I before deleting the edge junction 1691.

Reference is now made to Fig. 12 which is a simplified block diagram of the special points correspondence finding block of Figs. 3A and 4 - 8. A weight computation unit 1700 receives as input a list of special point candidates, typically from the ROI, and estimated or predicted special points and computes a correlation weight between each special point candidate and each estimated or predicted special point. The correlation weight is based on a correlation error. The estimated points may comprise known points from a previous frame. The operation of the weight computation unit 1700 is described in more detail below with reference to Fig. 13.

A threshold filter 1710 applies a minimum threshold to the weights received from the weight computation unit 1700 and outputs a thresholded weight. The threshold filter 1710 receives the correlation error from the weight computation unit 1700, and preferably computes an appropriate threshold based thereupon. A typical threshold is based directly on correlation error as, for example 0.125, when the correlation error is normalized in a range 0:1.

It is appreciated that the special point candidates may not necessarily come only from the ROI, but may also come from a region chosen based on distance from an estimated point, or from a region chosen based on other criteria. In such a case, the weight computation may take the distance into account.

Based on the thresholded weight an initial probability

is computed (unit 1720) for each candidate, showing its probability to be each of one or more special points. Typically, the initial probability for each point is computed as follows:

1. The possibility exists that no appropriate candidates exist. Therefore, a fictional candidate is added, preferably without a specific location, with an initial assigned probability of $Pr^* = (1 - W_{max})$,

where W_{max} = maximum correlation weight of all candidates.

2. For each candidate not including the fictional candidate, the probability is computed as follows:

$$Pr(j) = W_{max} * (W_j) / \text{SUM } (W_j),$$

where W_j is the weight of candidate j, and

SUM is taken over all candidates, not including the fictional candidate.

The candidates are then filtered in a candidates filter 1730, which picks the best possible candidate for each special point based on a filter criterion. The filter method may, for example, choose the candidate with the highest probability. Preferably, the method may use a more complex filtering criterion taking into account possible movements and irregularities in movement of the special points, and the relationships between them.

Reference is now made to Fig. 13, which is a simplified flowchart of a preferred method of operation for the special points weights computation unit 1700 of Fig. 12. An estimated or predicted special point and a special point candidate are input (steps 1810 and 1820). A correlation weight is computed (step 1830), based on the estimated or predicted special point and the special point candidate colors.

A preferred formula for computing correlation weight is as follows:

$$\text{correlation weight} = (1 / (1 + C * ER))$$

where C is a coefficient, preferably having a value of 10; and

ER is the normalized correlation error between a

special point candidate and an estimated/predicted special point.

A preferred formula for computing ER is as follows:

$$ER = \text{SQRT}((SI_1 + SI_2 + SI_3) / 27)$$

where SQRT is the square root;

$$\text{and } SI_n = \text{SUM}((I_n^K - I_n^{OK}) * (I_n^K - I_n^{OK})),$$

where SUM is the sum from k = 1 to k = 9,

I is the intensity of the pixel, normalized in the ranged 0:1,

K and OK are indexes representing a mask of pixels around the special point candidate and the estimated/predicted special point, respectively, and

n represents the index of the color, as defined above.

The above formulas are for a 3 x 3 mask. It is appreciated that masks of other sizes may also be used. If, for example, a 5 x 5 mask is used, the sum would be taken from 1 to 25 and the denominator in the formula for ER would be equal to 75 instead of 27.

The correlation computation is repeated for each combination of an estimated/predicted special point with all of its candidate points (steps 1850 and 1860).

Reference is now made to Fig. 14 which is a simplified flowchart of a preferred method of operation for the border estimation block of Figs. 3A and 4. Two consecutive corresponding special points are input (step 1910). An initial estimated border segment is input (1920). The initial estimated border segment connects the last two consecutive corresponding special points.

The estimation segment is projected between the consecutive corresponding special points, and an estimated border segment is created (step 1940).

The remaining special points are then processed until the last special point is reached (steps 1960 and 1980). The estimated border segments are then used to create an exact border description (step 1950).

Reference is now made to Fig. 15 which is a simplified flowchart of an alternative preferred method of operation for the

border estimation block of Figs. 3A and 4. In the method of Fig. 15, widths of ROIs (regions of interest) for borders are also computed. It is appreciated that the method of Fig. 15 is preferably performed at the conclusion of the method of Fig. 14. In the method of Fig. 15, a corresponding border segment is input (step 2010). The size of the ROI is selected as the size of the larger diameter of the two consecutive corresponding special points (step 2020).

Reference is now made to Fig. 16 which is a simplified flowchart of a preferred method of operation for the borders and mask generation unit of Figs. 3A and 4 - 8. In the method of Fig. 16, object special points and border segments are drawn according to the chain code description (step 2100). The term "drawn", as used in step 2100, does not necessarily indicate drawing in a visible form, but rather refers to creating an internal representation analogous to a drawn representation.

A seed-grow is created (step 2110) beginning from the frame of each picture. The seed-grow is limited by meeting an object special point or border segment, and does not go past a border segment. The seed-grow continues until no further growth is possible. Preferably, the seed grow begins on a portion of the picture frame which is not part of an object. Preferably, in order to ensure that the seed-grow begins on a portion of the picture frame which is not part of an object, an extra row of blank pixels is added all around the picture frame, and the seed-grow begins in one of the extra pixels.

Methods for seed-grow are well-known in the art, and are described in D. K. Ballard and C. M. Brown, Computer Vision, referred to above, at page 149.

Pixels of the picture are then assigned values (step 2130) as follows: area covered by the seed-grow, 0; other areas, 1; optionally, transition pixels, intermediate value between 0 and 1. Assigning an intermediate value to transition pixels may be preferred, for example, in the case where the mask being created includes anti-aliasing. Optionally, a border description may be created from the object mask by outputting only the

transition pixels.

It is appreciated that the method of Fig. 16 creates a particular output format consisting of a mask, and that many other output formats and data representations may be used as, for example, direct chain code output.

Reference is now made to Fig. 17 which is a simplified flowchart of a preferred method of operation for the exact object border description blocks of Figs. 3A, 4, 5, 6, 7 and 8. Fig. 17 comprises a method for operation of blocks 350, 450, 465, 550, 576, 650, 665, 750, and 850. The method of Fig. 17 also comprises a preferred method of operation for the border description portion of blocks 370, 470, 570, 670, 771, 774, 871, and 874.

The method of Fig. 17 preferably comprises the following steps.

In the case of blocks 350, 450, 550, 650, 750, and 850, it is appreciated that some steps of Fig. 17 may be omitted and only certain steps of Fig. 17, such as, for example, steps 2220 and 2230, are then performed. In this case, in place of corresponding special points and corresponding border segments, special points and border segments derived from the user's initial indication of the objects to be tracked may be input.

Edge thinning is performed on border segments, preserving special points (step 2202). Along the direction of the edge, if the edge is greater than one pixel in width, the width is reduced to one pixel. The reduction to one pixel is accomplished by keeping only one pixel, either the central pixel in the case where the edge is an odd number of pixels in width, or one of the two central pixels if the edge is an even number of pixels in width. However, every pixel constituting a special point is kept, and if the special point is not a central pixel then the special point is kept and other pixels are not kept.

The thinned border segments from step 2202 are merged with the edge picture of the border ROI (step 2204).

The merged output of 2204 is thinned again in step 2206, similarly to the thinning of step 2202 described above.

A new component map is created from the output of step 2206, preferably only including the special points and border segments (step 2210). A preferred method for performing step 2210 is similar to that described above with reference to Figs. 10 and 11.

A seed-grow is performed (step 2220), similar to the seed-grow described above with reference to step 2110 of Fig. 16. The grow limit for the seed grow of step 2220 is any border segment.

In step 2230, a chain code is computed from the new component map as follows. Elements bounded by the seed-grow area are marked as external elements. "Bounded" is understood in step 2230 to mean surrounded, except possibly at junction points. Elements not touched at all by, that is, not bordering at all on the seed-grow area, are marked as internal elements. Other elements, touched but not bounded by the seed-grow area are marked as border elements.

A chain code is then computed, linking, in order, all of the border elements. For further processing, the chain code is taken to comprise an exact border description, and the junctions described therein are taken as estimated special points.

Reference is now made to Fig. 18 which is a simplified flowchart of a preferred method of operation of the following elements: 570, 572, and 574 of Fig. 5, combined; 670, 672, and 674 of Fig. 6, combined; 771, 772, 774 of Fig. 7, combined; and 871, 872, 874 of Fig. 8, combined. It is appreciated that, in the case of elements 774, and 874, Fig. 18 describes only a portion of the operation thereof and does not include other portions which are described above with reference to Fig. 17.

The steps of the method of Fig. 18 are similar to those of Fig. 14, and are self-explanatory with reference to the above description of Fig. 14, except as described below.

In step 2340, border correspondences are found, including compensating for border segments not found by using the

estimated border segments.

In step 2370, an exact border description is created from all corresponding segments.

Reference is now made to Fig. 19 which is a simplified flowchart of an alternative method of operation of step 2340 of Fig. 18.

In step 2410, a distance map is built. A distance map is a map which indicates the distance of each pixel from an individual estimated border segment. Preferably, the distance map is built for each estimated border segment within the ROI. In the case that two end corresponding points have a different ROI size, the larger ROI size is preferably used.

Preferably, the distance map is created as follows:

a. each border segment pixel is assigned a distance of 0;

b. each unassigned pixel adjacent to a pixel that was already assigned a distance of n is assigned a distance of $n+1$, except for pixels diagonally adjacent to the last pixel at the end of the region of assigned pixels, which is not assigned a distance;

c. step b is repeated until each pixel within the ROI has been assigned a distance.

In step 2420, color parameters of border candidates are computed. Border candidates are typically edges found in the ROI. The color parameters preferably comprise average and tolerance. Preferably, each of the average and the tolerance are computed separately for a strip, typically of width 1 pixel, adjacent to the edge at the interior and exterior thereof. The interior and exterior are distinguished based on the direction of traversal of the border.

The average is computed as separate averages for each color component, each taken to be the average value of one of the three color components I_1 , I_2 , and I_3 . The tolerance, also computed separately for each color component, describes the tolerance of the average color, and is typically based on the variance.

Step 2425 is similar to step 2420, except that the input to step 2425 is a found border segment.

In step 2430, a weight is computed for each border segment, representing the similarity between the candidate border segments and the found border segment. Typically, separate weights are computed based on average color, average distance, tolerance of average color, and tolerance of distance. Typically, average distance is computed based on the average of the distances assigned to the pixels in the candidate border segment by the distance map computed previously, as described in step 2410.

In step 2440 a threshold filter applies a minimum threshold to the weights received from the weight computation 2430 and outputs a combined, thresholded weight.

Based on the combined, thresholded weight an initial probability is computed (unit 2450) for each candidate, representing its probability to be a part of a border segment corresponding to the found border segment. The candidates are then filtered in a candidates filter 2460, which picks the best possible group of one or more candidates for the border segment corresponding to the found border segment, based on a filter criterion. Preferably, the filter method takes into account the probability that each candidate is part of the border segment, as well as the relationship, that is, distance and angle, between the candidates with respect to the border segment. The filter method may employ maximum probability methods or any appropriate statistical iteration method.

Parts of the border segment that were not found previously in step 2460 are filled in step 2470 using the estimated border segment or parts thereof.

Reference is now made to Fig. 20 which is a simplified flowchart of a prediction method useful in the methods of Figs. 7 - 8. The method of Fig. 20 is particularly useful for steps 752, 777, 852, and 866. The method of Fig. 20 preferably comprises the following steps.

The method of Fig. 20 refers to the case in which

frame-by-frame processing is occurring. A check is made for whether chain code is available for four or more consecutive frames (step 2810). If not, processing continues with step 2820, described below.

A third order prediction of borders and/or special points is performed (step 2815). Step 2815 is described in more detail in Fig. 23, below, for the case of special points. Typically, borders are predicted similarly by using equations of motion on the control points of the splines of the estimated border segments.

Similarly, in steps 2820 and 2830 a decision is made, based on how many chain codes are available, as to whether a second order prediction (step 2825) or a first order prediction (step 2835) is to be made. Step 2825 is described more fully in Fig. 22 below, and step 2835 in Fig. 21 below.

In step 2840, sufficient information is not available for a prediction. In this case, the user is asked to identify the desired object in the second frame (step 2845) and, if necessary, in the first frame (step 2850).

Reference is now made to Fig. 21 which is a simplified flowchart of a preferred method for carrying out the steps of Fig. 20 in the case of first-order prediction. The method of Fig. 21 preferably comprises the following steps.

First frame and second frame corresponding special points are input. Points found only in the second frame are added to the first frame (step 2910), so that the same number of points will be found in each frame, allowing a complete prediction for the next frame. In step 2910, point location is preferably determined by reverse geometrical interpolation along the chain-code edge, according to the points' location relative to the location of the two chain-code points, found in both frames, which bound the point to be interpolated.

In step 2920, the special point velocities, next frame first order predicted location, and next frame special point ROI size are computed.

Reference is now made to Fig. 22 which is a simplified

flowchart of a preferred method for carrying out the steps of Fig. 20 in the case of second-order prediction. The method of Fig. 22 is self-explanatory with reference to the above description of Fig. 21, step 3010 being analogous to step 2910, and step 3020 being analogous to step 2920.

In step 3020, it is appreciated that the computation of next frame special point ROI size is optional. Alternatively, the current frame ROI size may be used.

Reference is now made to Fig. 23 which is a simplified flowchart of a preferred method for carrying out the steps of Fig. 20 in the case of third-order and higher prediction. The method of Fig. 23 is self-explanatory with reference to the above description of Figs. 21 and 22, step 3110 being analogous to steps 2910 and 3010, and step 3120 being analogous to step 2920 and 3020.

In step 3105, a decision is made, based on how many frames have been previously processed, as to how many frames are used in subsequent steps. In step 3120, it is appreciated that the computation of next frame special point ROI size is optional. Alternatively, the current frame ROI size may be used.

Referring back to Fig. 1, another example of a suitable application device is an effect device 92 such as, for example, a device for performing one or more of the following effects: compression; painting; blurring; sharpening; a filter operation; and an effect which changes over time at a different rate on different sides of the border.

Optionally, the application devices 80, 90 or 92 operate on an individual frame, and cause the result of this operation to be displayed to the user, before operation proceeds generated for a subsequent frame.

Typically, the video display apparatus 95 or a separate video display apparatus (not shown) is employed to display a result of performing the operation without previously displaying a separate representation of said border.

A result of performing an operation, without previously displaying a separate representation of the border, may be dis-

played for a plurality of frames rather than for a single frame.

An advantage of this option is that interactive correction may be effected not on the basis of viewing a representation of the border but rather on the basis of viewing an effect or application generated by the application devices assuming a particular border. Viewing an effect or application is often a more useful method for evaluating the quality of the border tracking, relative to viewing a representation of the border itself as tracked.

Preferably, the result of performing an operation such as an effect or application is displayed together with a representation of the border as tracked. If the result of performing the operation is deemed by the user unsatisfactory, the user uses the display to correct the border. Preferably, the display changes automatically to reflect the change in the result of performing the operation due to the new border.

Referring back to Fig. 2A, blocks 135 and 165 indicate that, optionally, an effect or application is carried out, e.g. by an external application device. Preferably, the application or effect is carried out before the user is prompted to decide whether or not the border has been appropriately tracked (steps 140, 170). Thereby, the user can employ the results of the application or effect to evaluate the border as tracked.

Referring back to Fig. 2G, blocks 235 and 265 indicate that, optionally, an effect or application is carried out, e.g. by an external application device. Preferably, the application or effect is carried out before the user is prompted to decide whether or not the border has been appropriately tracked (steps 240, 270). Thereby, the user can employ the results of the application or effect to evaluate the border as tracked.

Reference is now made to Fig. 24 which is similar to Fig. 4 except for the following differences, which can exist either separately or in combination:

- a. Subsequent frames are only brought (step 455) once the user has deemed the border, as tracked, satisfactory, in the current frame.

b. The user determines whether or not the border is satisfactory by reviewing results of operations (effects and/or applications) generated with the assumption that the border's location in the current frame is as tracked.

Due to these differences, step 490 in Fig. 4 is replaced with step 492 in Fig. 24.

It is appreciated that one or both of the above modifications may also be effected on the apparatus of Fig. 6.

Reference is now made to Fig. 25 which is similar to Fig. 8 except for the following differences, which can exist either separately or in combination:

a. Special points are only predicted in a subsequent frame (step 866) once the user has deemed the border, as tracked, satisfactory, in the current frame.

b. The user determines whether or not the border is satisfactory by reviewing results of operations (effects and/or applications) generated with the assumption that the border's location in the current frame is as tracked.

Due to these differences, step 890 in Fig. 8 is replaced with step 892 in Fig. 25.

Reference is now made to Fig. 26 which is similar to Fig. 3A except for the following differences, which can exist either separately or in combination:

a. Other frames are only brought (step 355) once the user has deemed the border, as tracked, satisfactory, in the current frame.

b. The user determines whether or not the border is satisfactory by reviewing results of operations (effects and/or applications) generated with the assumption that the border's location in the current frame is as tracked.

Due to these differences, step 390 in Fig. 3A is replaced with step 392 in Fig. 26.

It is appreciated that one or both of the above modifications may also be effected on the apparatus of Fig. 5.

Reference is now made to Fig. 27 which is similar to Fig. 7 except for the following differences, which can exist

either separately or in combination:

- a. Special points are only predicted in another frame or frames (step 777) once the user has deemed the border, as tracked, satisfactory, in the current frame.
- b. The user determines whether or not the border is satisfactory by reviewing results of operations (effects and/or applications) generated with the assumption that the border's location in the current frame is as tracked.

Due to these differences, step 790 in Fig. 7 is replaced with step 792 in Fig. 27.

The keyframe examination blocks (335 in Fig. 3A, 435 in Fig. 4, 535 in Fig. 5, 635 in Fig. 6, 735 in Fig. 7, 835 in Fig. 8) and other examination blocks (379 in Fig. 3A, 478 in Fig. 4, 578 in Fig. 5, 678 in Fig. 6, 778 in Fig. 7, 878 in Fig. 8, 435 and 478 in Fig. 24, 835 and 878 in Fig. 25, 335 and 379 in Fig. 26 and 735 and 778 in Fig. 27) have two possible outcomes: if, as is preferred, the user performs an examination of the border and/or results of operating differentially on both sides of the border in the current frame and deems it unsatisfactory, the border in the current frame is typically corrected, preferably in response to a user-indicated correction. If the user deems the current frame satisfactory, the method proceeds to other frames.

In hotspot applications or other applications in which the location of the dynamic object is presented to the user, the mask generated in step 2130 of Fig. 16 may be used to determine whether the user is pointing at a dynamic object which may, for example, be a hotspot, or whether the user is pointing at a background location which is not a hotspot.

It is appreciated that, while certain components of the present invention are described above with reference to processing of color image sequences, the present invention is not limited to the processing of color images but may also process, for example, monochrome and grayscale images.

It is appreciated that the software components of the present invention may, if desired, be implemented in ROM (read-only memory) form. The software components may, generally, be

implemented in hardware, if desired, using conventional techniques.

It is appreciated that various features of the invention which are, for clarity, described in the contexts of separate embodiments may also be provided in combination in a single embodiment. Conversely, various features of the invention which are, for brevity, described in the context of a single embodiment may also be provided separately or in any suitable subcombination.

It will be appreciated by persons skilled in the art that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention is defined only by the claims that follow:

CLAIMS

1. A tracking method comprising:
receiving a representation of an event including at least one dynamic object having a border and having at least one edge portion which is absent during at least a portion of the event; and
providing an ongoing indication of the location of the border of the object during the event.
2. A method according to claim 1 wherein said representation comprises a video representation.
3. A method according to claim 1 or claim 2 wherein said edge portion comprises a portion of said border.
4. A method according to any of claims 1 - 3 and also comprising reconstructing at least one absent edge portion.
5. A tracking method comprising:
receiving a representation of an event including at least one dynamic object having a border; and
providing an ongoing indication of the location of the border of the object during the event.
6. An edge-tracking method for tracking at least one dynamic object appearing in a sequence of frames, the method comprising:
for at least one key frame within the sequence of frames, marking at least one edge of at least one dynamic object based at least partly on external input; and
for all frames within the sequence of frames other than said at least one key frame, automatically marking at least one edge of at least one dynamic object based on output from the first marking step.
7. A method according to claim 6 and also comprising remarking said at least one automatically marked edge at least

once, based on external input.

8. A method according to claim 6 or claim 7 wherein said external input comprises human operator input.

9. A method according to any of claims 6 - 8 wherein at least one edge is marked without detecting the edge.

10. A method according to any of claims 6 - 9 wherein said at least one key frame comprises a subsequence of frames preceding all other frames within the sequence.

11. A method according to any of claims 6 - 9 wherein said at least one key frame comprises a subsequence of frames following all other frames within the sequence.

12. A method according to any of claims 6 - 9 wherein said at least one key frame comprises a subsequence of frames preceding at least one other frame within the sequence and following at least one other frame within the sequence.

13. An edge-structuring method for structuring a plurality of connected edges into a graph, the method comprising:

providing a plurality of connected edges;

traversing the plurality of connected edges in a chosen direction; and

structuring the plurality of connected edges into a graph comprising a branch list and a node list,

wherein the node list is independent of the chosen direction.

14. A method according to claim 13 wherein the node list comprises an edge junction list.

15. A method according to claim 13 or claim 14 wherein the node list comprises an edge terminal point list.

16. A method according to any of claims 13 - 15 wherein the node list comprises an edge corner list.

17. A method according to any of claims 13 - 16 wherein the node list comprises a curvature list.

18. A method according to any of claims 13 - 17 wherein the plurality of connected edges comprises a plurality of pixels and wherein the traversing step comprises:

specifying a current pixel;

identifying at least one visible pixel associated with the current pixel; and

classifying the current pixel based, at least in part, on the number of visible pixels identified.

19. A method according to claim 18 wherein the identifying step includes:

defining a blind strip; and

ruling out as visible pixels at least one pixel associated with the blind strip.

20. A method according to claim 19 wherein the ruling out step comprises ruling out as visible pixels all pixels associated with the blind strip whenever there is at least one visible pixel not associated with the blind strip.

21. A method for tracking a border of a moving object, the method comprising:

selecting a plurality of border locations to be tracked in a first image;

tracking at least some of said plurality of border locations from the first image to a second image; and

computing said border in said second image based on an output of said tracking step and based on information

characterizing said border in said first image.

22. A method according to claim 21 wherein at least one of said plurality of border locations comprises a location at which at least one border characteristic changes.

23. A method according to claim 22 wherein said border characteristic comprises at least one color adjacent to said border.

24. A method according to any of claims 21 - 23 wherein said tracking comprises disregarding a border location which, when tracked from said first image to said second image, is found to have moved differently from other adjacent border locations.

25. A method according to any of claims 21 - 24 wherein said computing step comprises transforming the border in the first image such that each of said plurality of border locations in said first image is transformed onto a corresponding one of the plurality of border locations in the second image.

26. A method according to any of claims 21 - 25 and also comprising identifying an actual border in said second image by searching adjacent to said border as computed in said second image.

27. A method according to claim 26 wherein an actual border is identified depending on whether the adjacent colors of the actual border resemble the adjacent colors of the border in said first image.

28. A method according to claim 26 or claim 27 wherein an output border is defined as said actual border, if identified, and as said border as computed in said second image, if no actual border is identified.

29. A method according to claim 26 or claim 27 wherein a first output border is defined which coincides in part with said actual border, where said actual border has been identified, and in part with said border as computed in said second image, where said actual border has not been identified.

30. A method according to claim 29 and also comprising:
identifying a new actual border in said second image by searching adjacent to said first output border; and
defining a new output border which coincides in part with said new actual border, where said new actual border has been identified, and in part with said first output border, where said new actual border has not been identified.

31. A method according to claim 25 wherein said transforming step comprises transforming a spline representation of the border in the first image such that each of said plurality of border locations in said first image is transformed onto a corresponding one of the plurality of border locations in the second image.

32. A method according to claim 21 and also comprising providing a first image seen from a first field of view and providing a second image seen from a different field of view.

33. A method according to claim 21 and also comprising providing first and second images each including at least one of a moving dynamic object and a dynamic background.

34. An image marking method including:
receiving a representation of an event, the representation including a plurality of frames, the event including at least one dynamic object having a border;
computing the location of the border of the dynamic object during the event; and

providing a user-sensible indication of locations of the dynamic object during the event, without previously displaying a separate representation of said border.

35. A method according to claims 6 - 12 wherein said automatic marking step comprises automatically marking all edges of at least one dynamic object based on output from the first marking step.

36. Tracking apparatus comprising:

event input apparatus operative to receive a representation of an event including at least one dynamic object having a border and having at least one edge portion which is absent during at least a portion of the event; and

a border locator operative to provide an ongoing indication of the location of the border of the object during the event.

37. Edge-tracking apparatus for tracking at least one dynamic object appearing in a sequence of frames, the apparatus comprising:

an edge marker operative, for at least one key frame within the sequence of frames, to mark at least one edge of at least one dynamic object based at least partly on external input; and

an automatic edge marker operative, for all frames within the sequence of frames other than said at least one key frame, to automatically mark at least one edge of at least one dynamic object based on output from the first marking step.

38. Edge-structuring apparatus for structuring a plurality of connected edges into a graph, the apparatus comprising:

an edge traverser operative to traverse the plurality of connected edges in a chosen direction; and

a graph structurer operative to structure the plurality

of connected edges into a graph comprising a branch list and a node list,

wherein the node list is independent of the chosen direction.

39. Apparatus for tracking a border of a moving object, the apparatus comprising:

a border selector operative to select a plurality of border locations to be tracked in a first image;

a border tracker operative to track at least some of said plurality of border locations from the first image to a second image; and

border computation apparatus operative to compute said border in said second image based on an output of said border tracker and based on information characterizing said border in said first image.

40. Tracking apparatus comprising:

event input apparatus operative to receive a representation of an event including at least one dynamic object having a border; and

a border locator operative to provide an ongoing indication of the location of the border of the object during the event.

41. A method according to any of claims 1 - 5, 21 - 33 and also including generating an effect which is applied differentially on different sides of said border.

42. A method according to any of claims 6 - 12 and 35 and also including generating an effect which is applied differentially on different sides of said at least one edge.

43. A method according to claim 41 or claim 42 wherein said effect includes an effect which is carried out at a location determined by a portion of said dynamic object.

44. An image modification method including:

receiving a representation of an event, the representation including a plurality of frames, the event including at least one dynamic object having a border;

computing the location of the border of the dynamic object during the event;

generating an effect which is applied differentially on different sides of said border; and

displaying a result of applying said effect without previously displaying a separate representation of said border.

45. A method according to any of claims 41 - 44 and wherein said step of generating an effect is performed on a subsequence of frames, including a plurality of frames, within said sequence of frames after an automatic marking step has been performed for said subsequence of frames.

46. A method according to any of claims 41 - 44 and wherein said step of generating an effect is performed on an individual frame from among said sequence of frames after an automatic marking step has been performed for said individual frame.

47. A method according to any of claims 41 - 44 wherein said effect is generated and displayed for an individual frame before the effect is generated for a subsequent frame.

48. A method according to any of claims 41 - 44 wherein said effect is displayed for all of said plurality of individual frames without expecting user input between frames.

49. A method according to any of claims 41 - 48 wherein said effect includes one of the following group of effects:

compositing;

retouching;

smoothing;

compression;

compositing;

painting;

blurring;

sharpening;
a filter operation; and
an effect which changes over time at different rates
on different sides of said border or said edge.

50. A method according to claim 34 wherein said event includes a plurality of dynamic hotspot objects and wherein said providing step includes providing an ongoing indication of locations of borders of each of the plurality of dynamic hotspot objects during the event.

51. A method according to claim 50, the method also including the steps of:

using said ongoing indication of locations of the borders of each of said hotspot objects to interpret a user's selection of an individual one of said plurality of dynamic hotspot objects; and

displaying information regarding the individual dynamic hotspot object selected by the user.

52. A method according to any of claims 34, 50 and 51 wherein said dynamic object is a portion of a larger object.

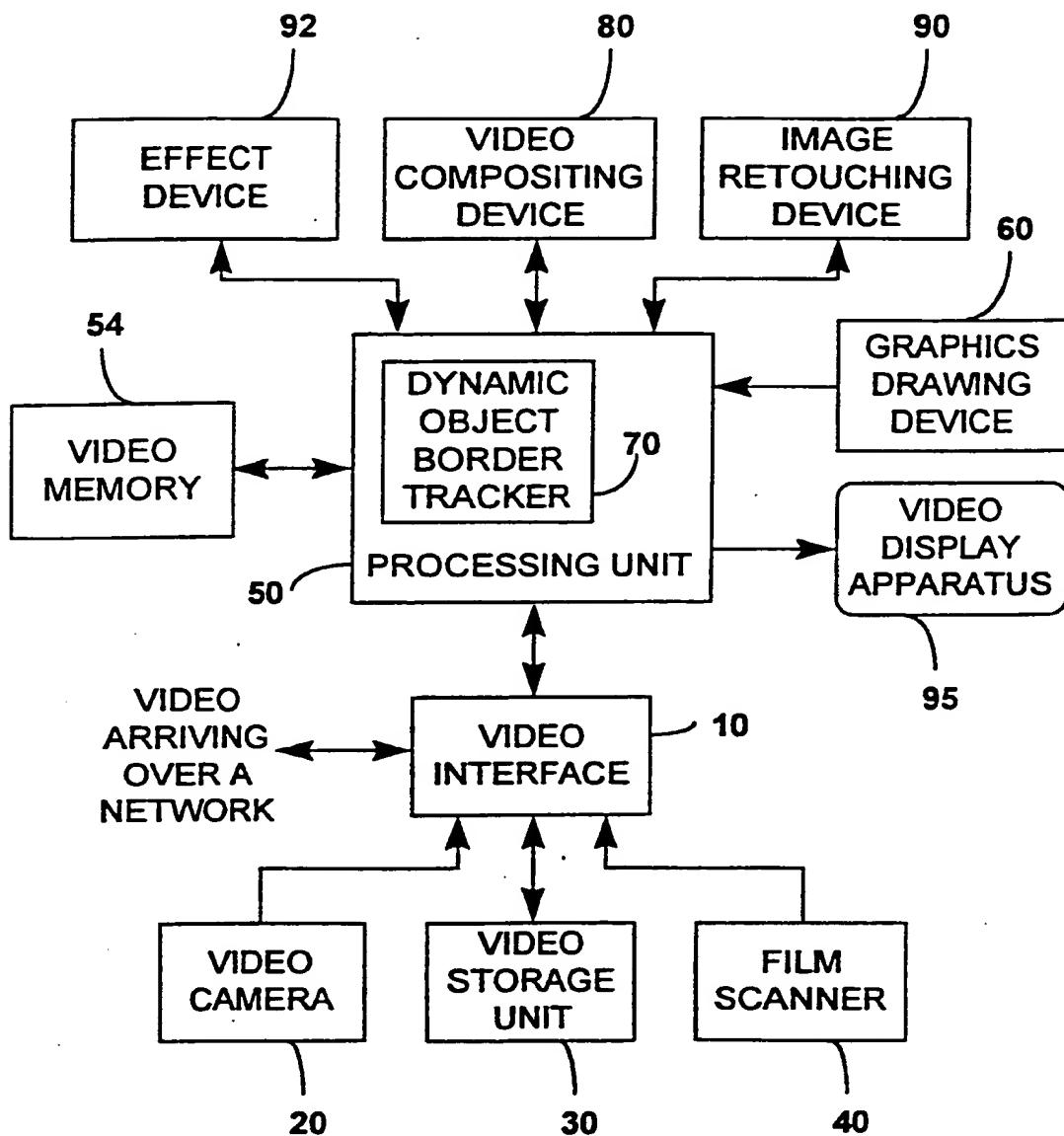
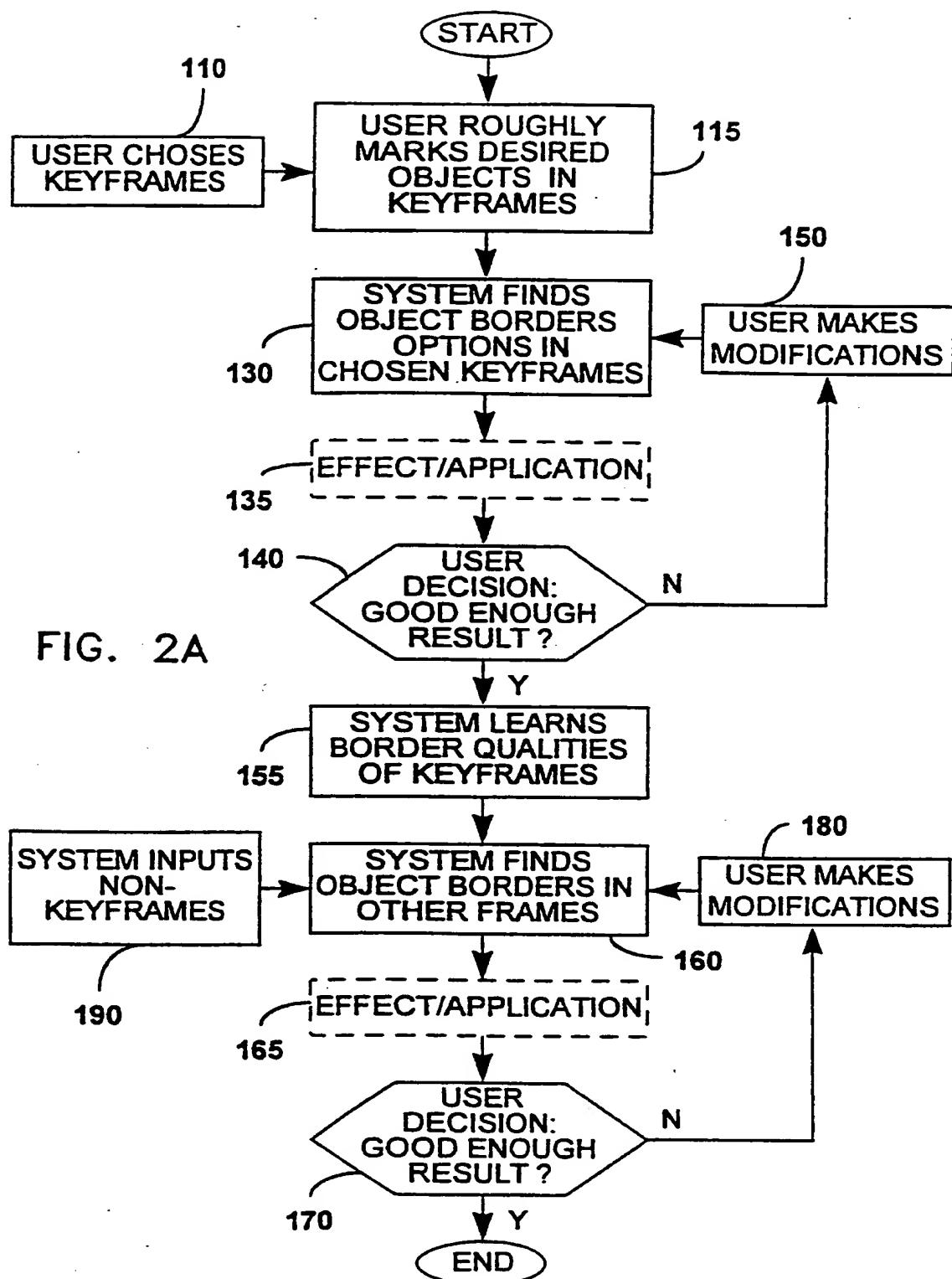


FIG. 1



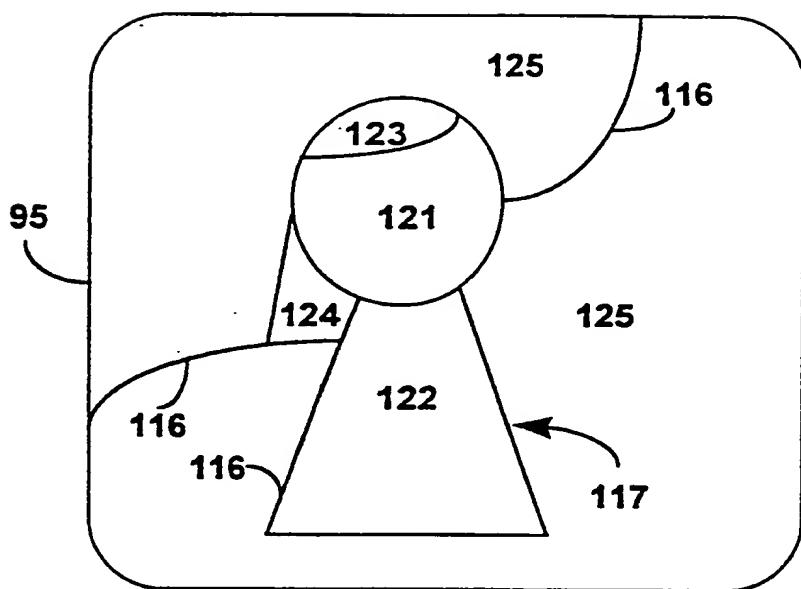
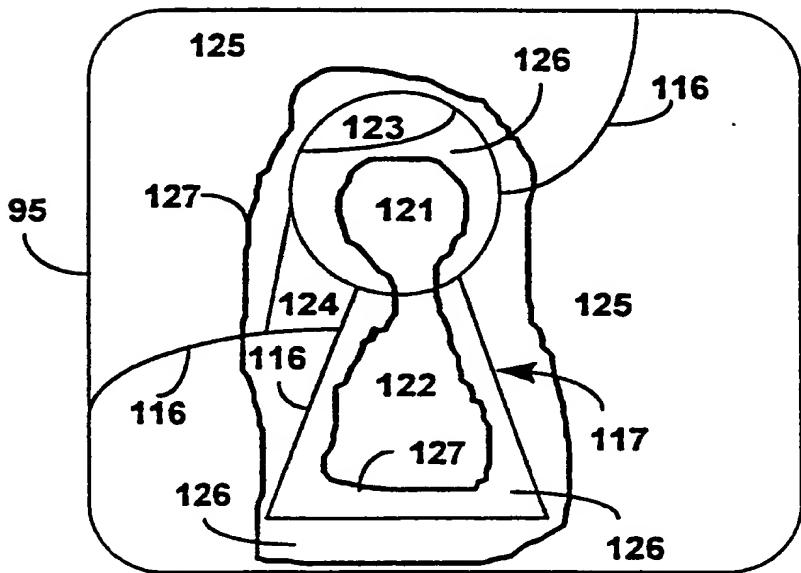


FIG. 2B

FIG. 2C



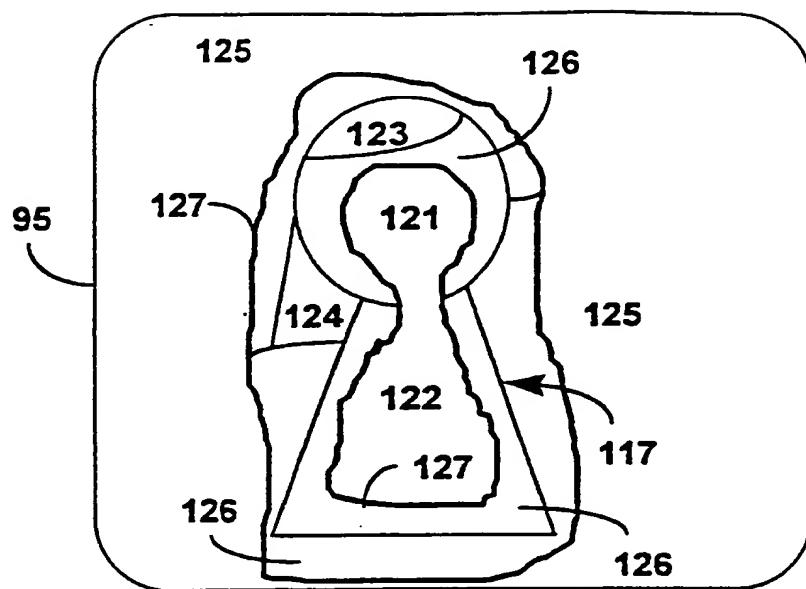
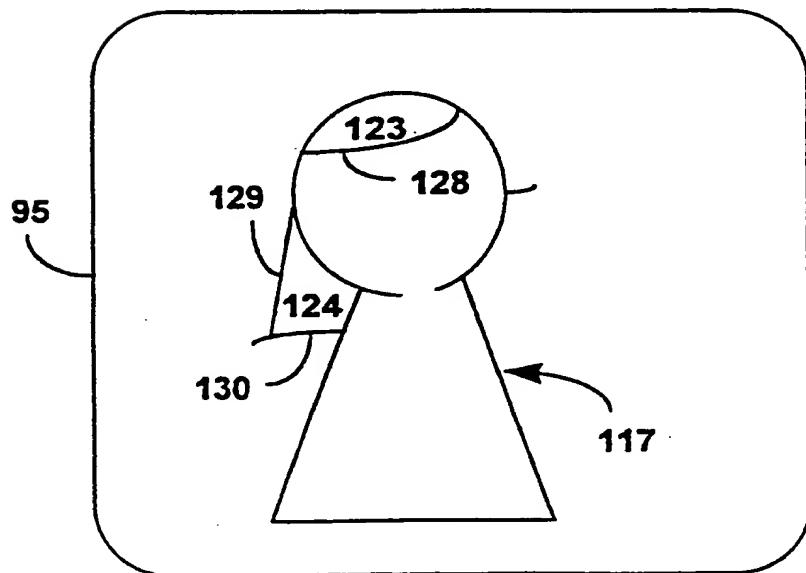


FIG. 2D

FIG. 2E



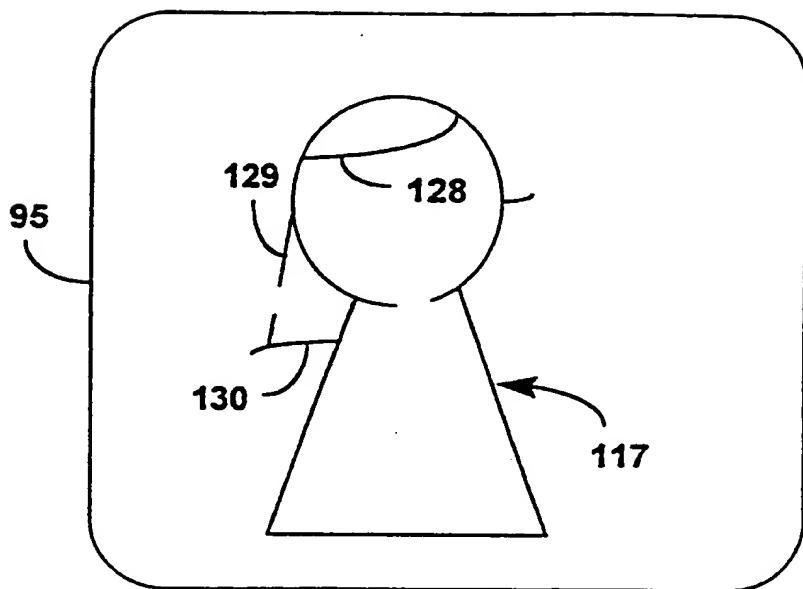
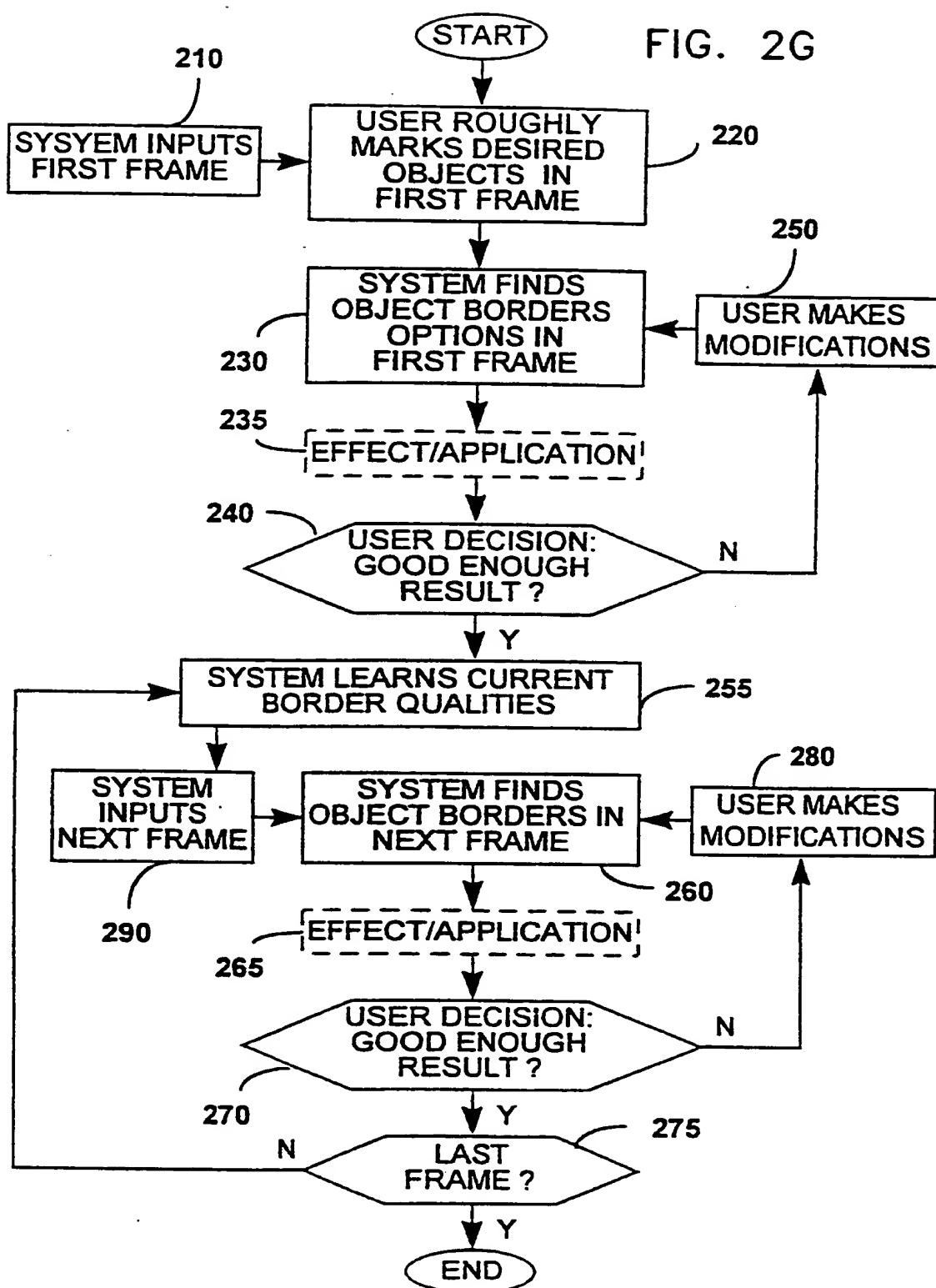


FIG. 2F

6/51

FIG. 2G



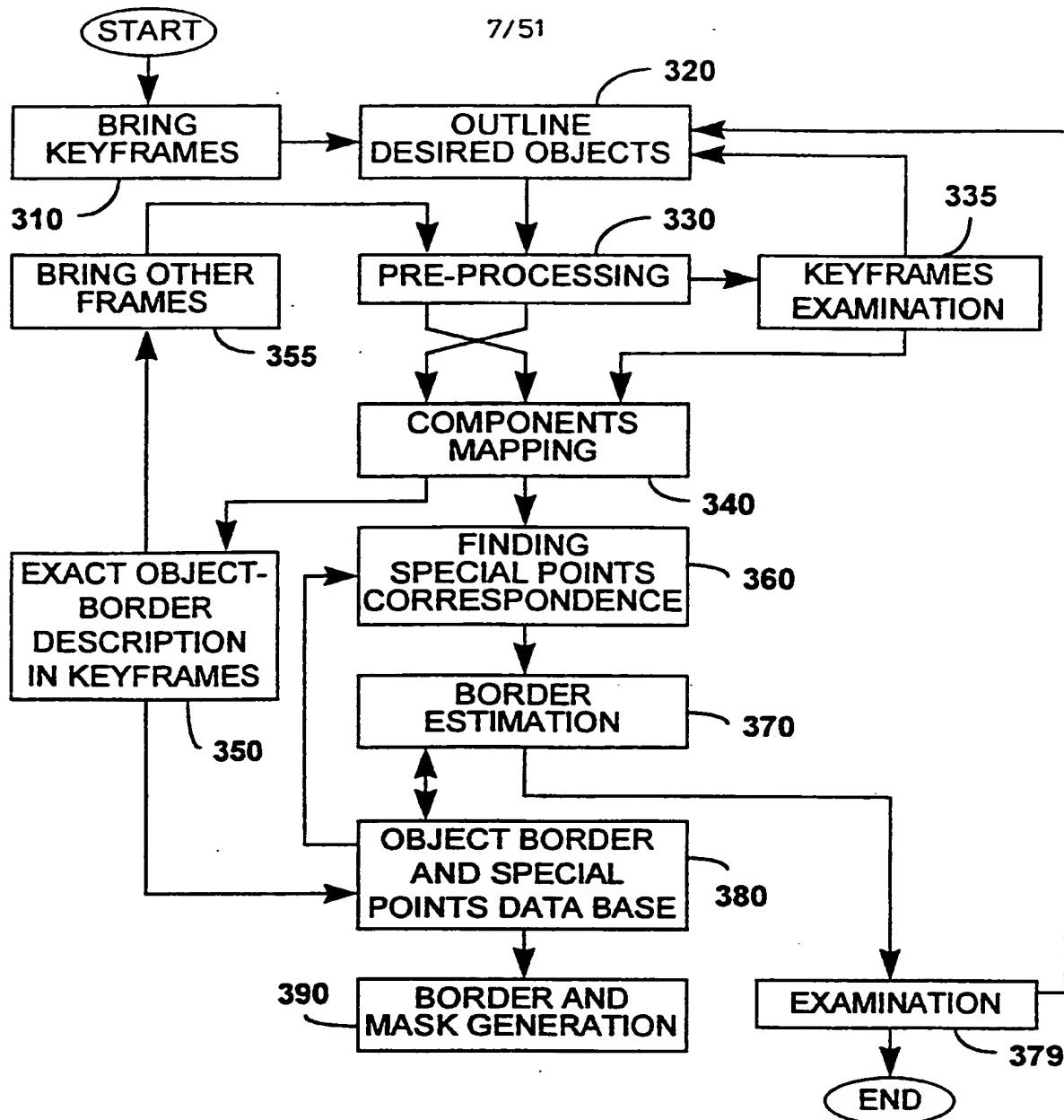


FIG. 3A

8/51

FIG. 3B

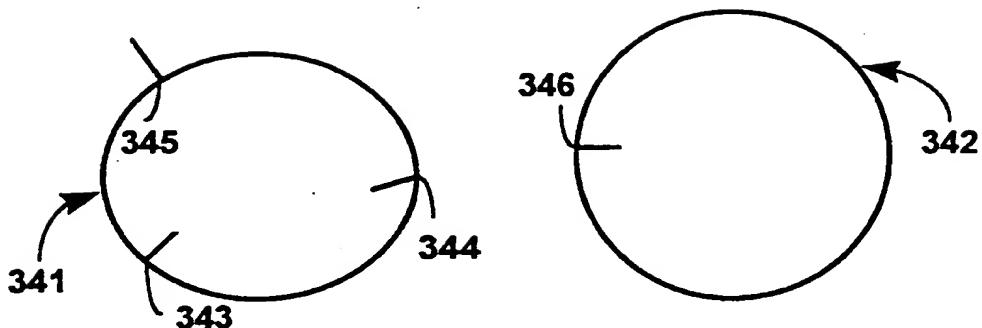


FIG. 3C

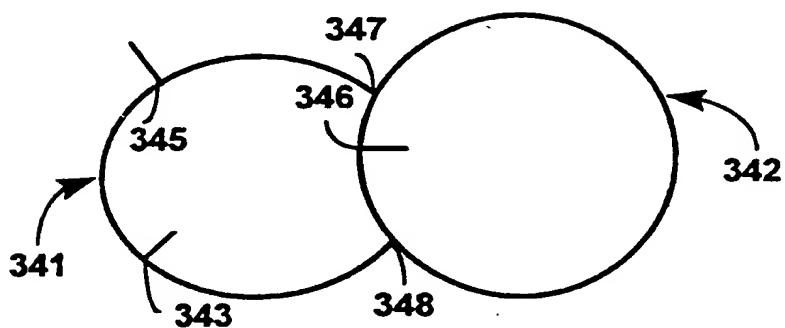
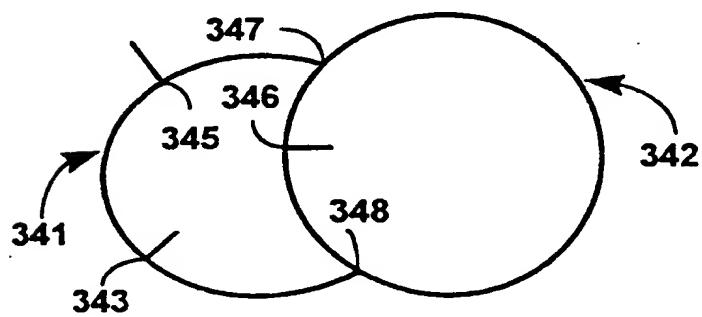


FIG. 3D



9/51

FIG. 3E

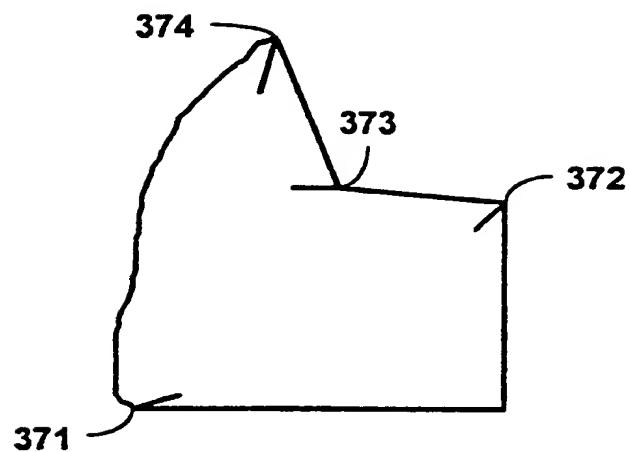


FIG. 3F

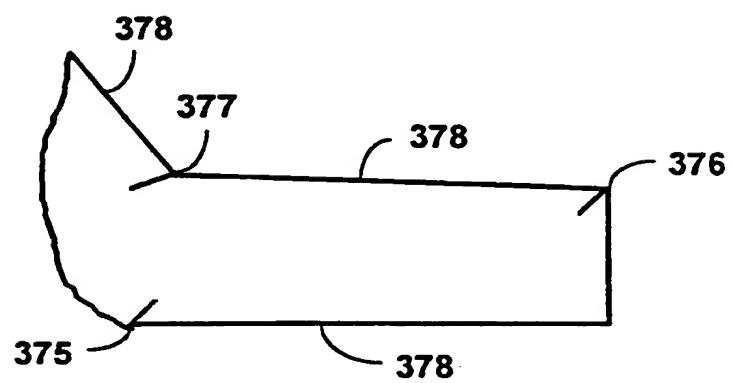


FIG. 4

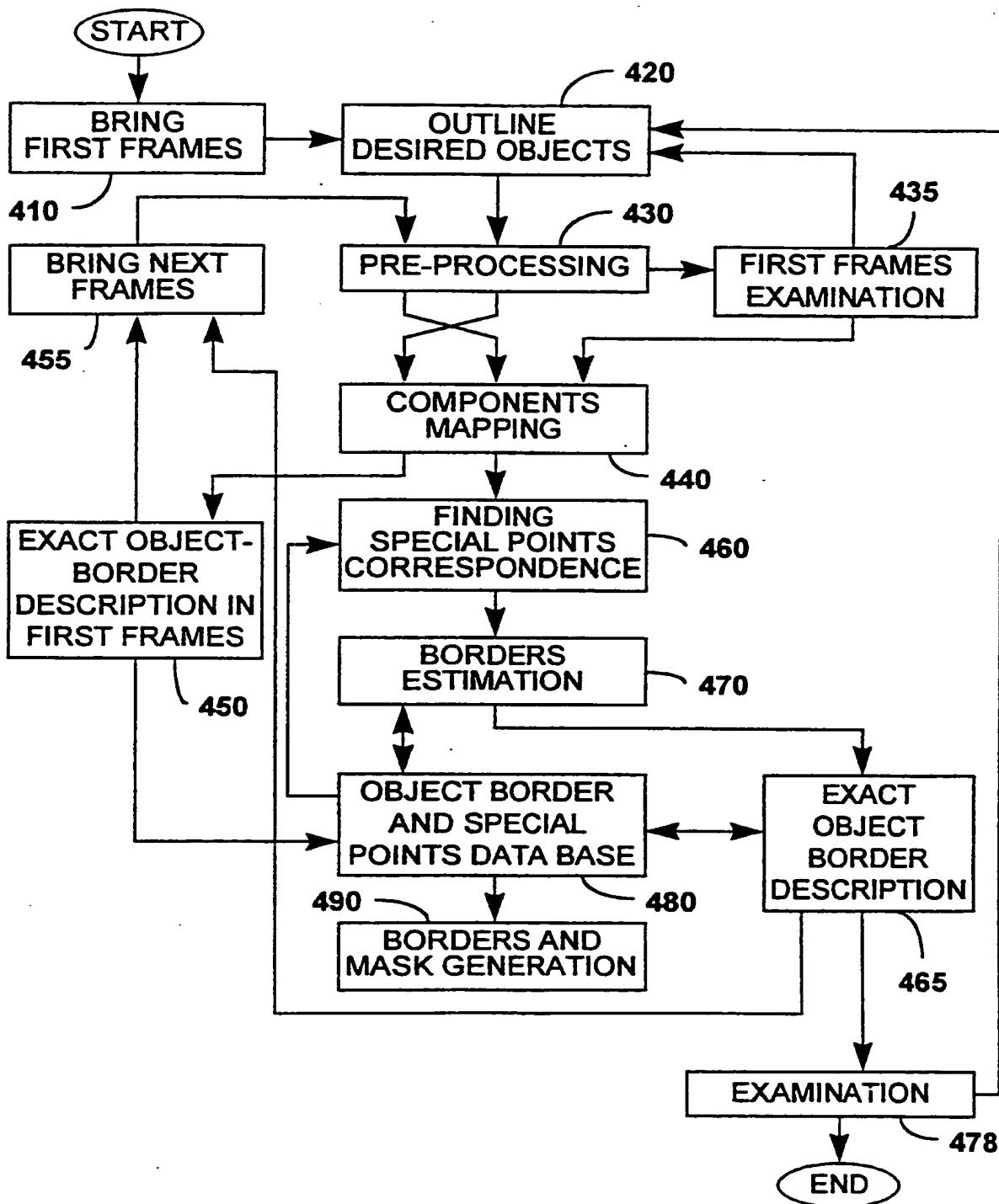


FIG. 5

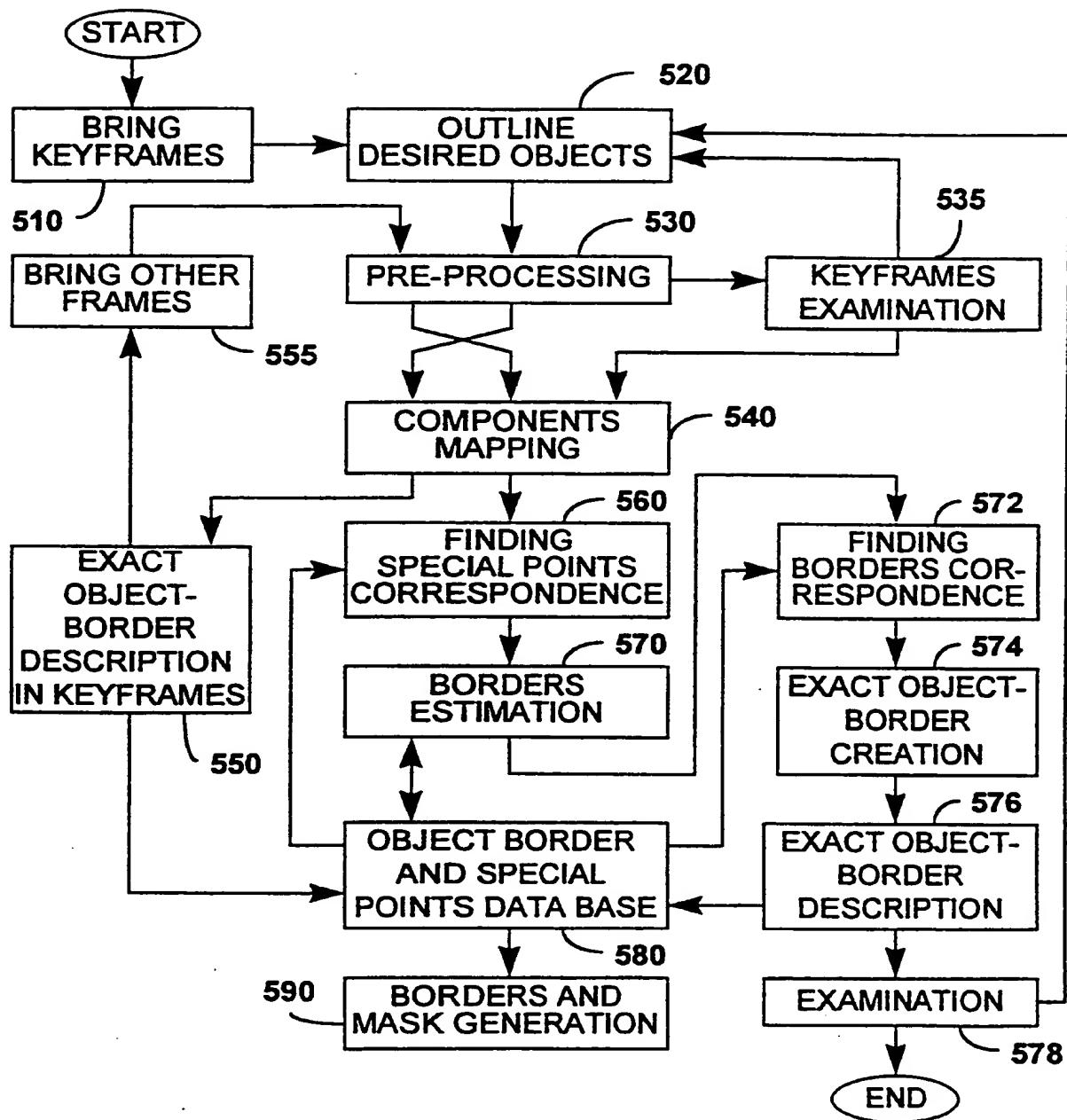


FIG. 6

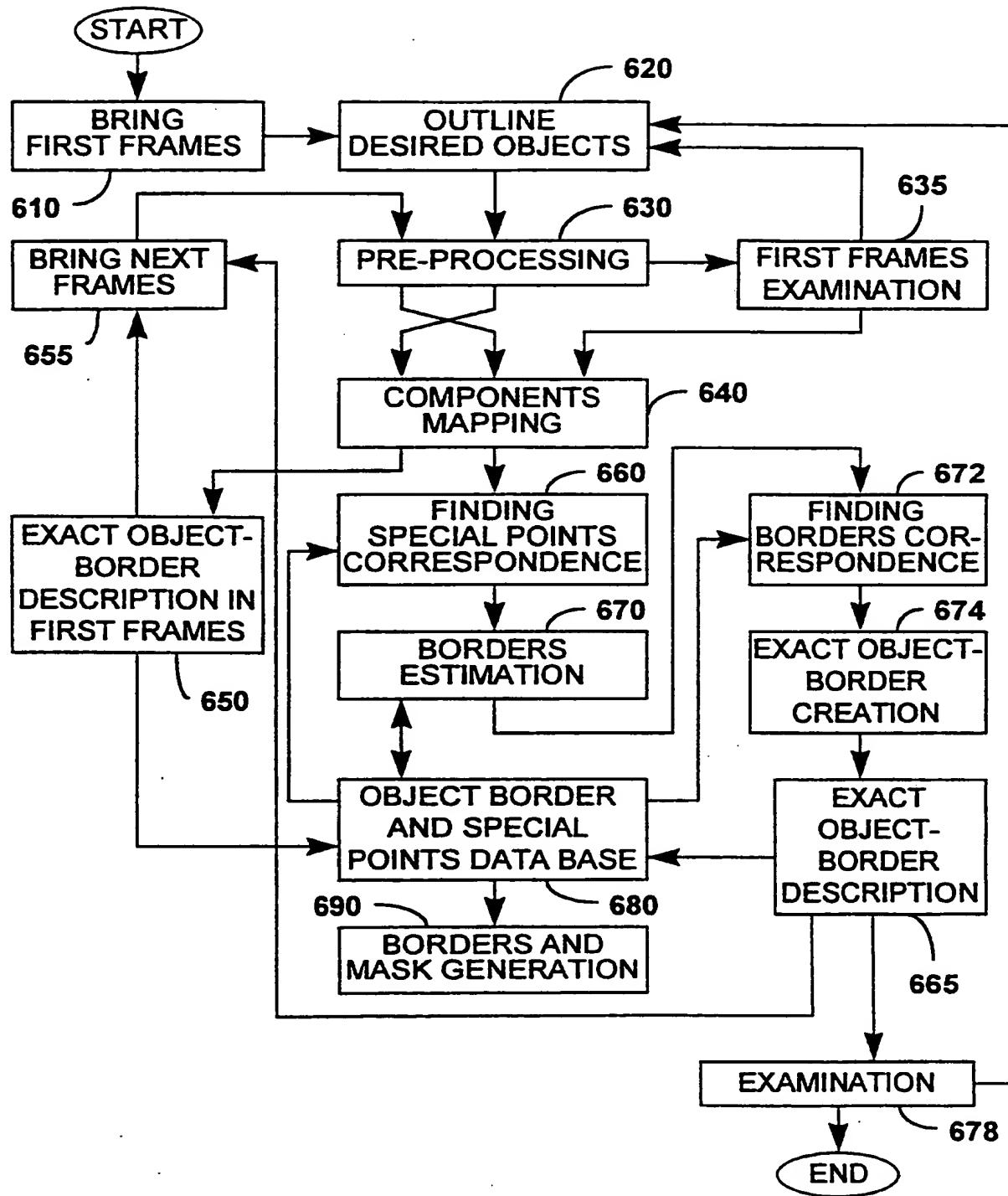


FIG. 7

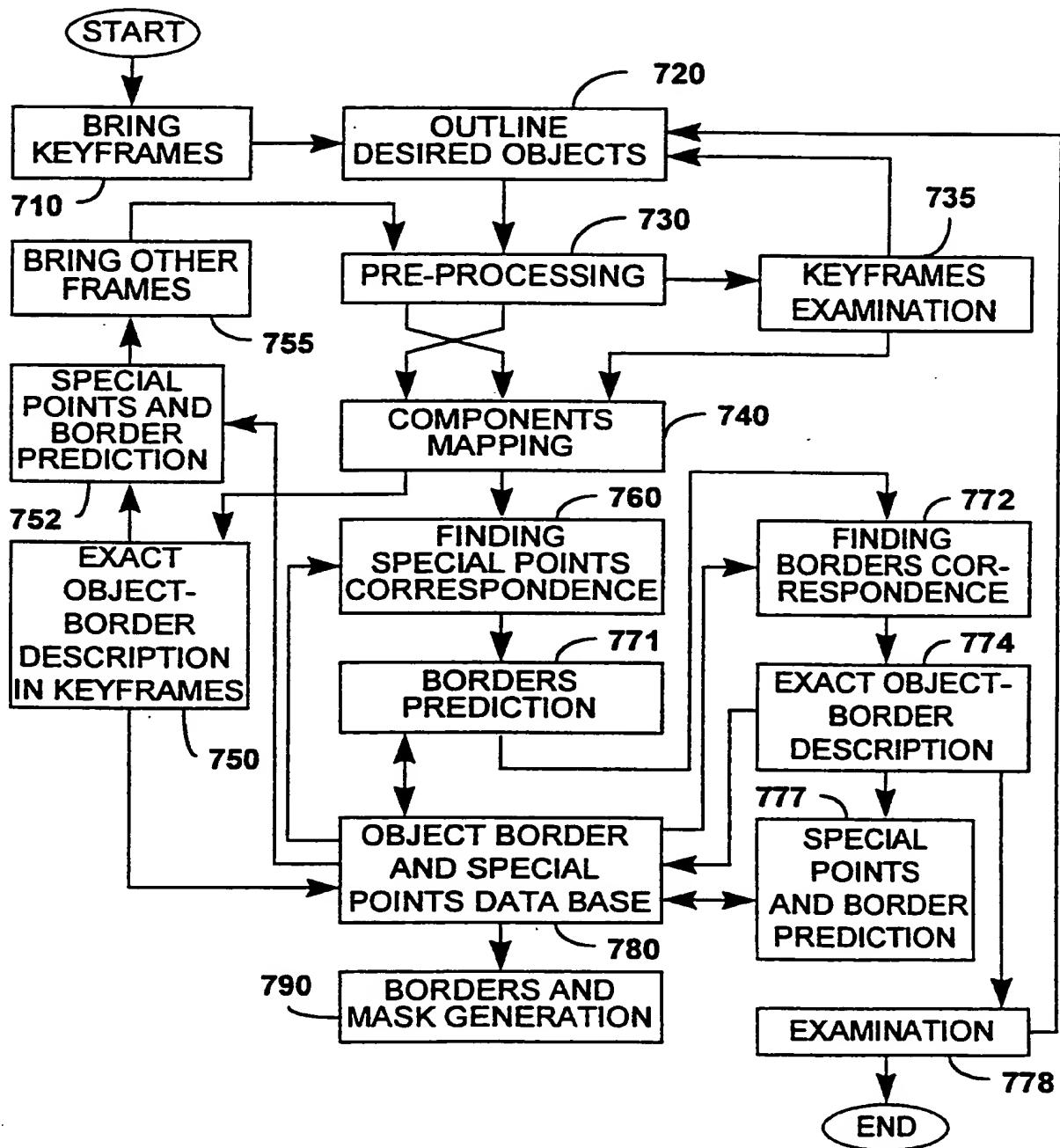


FIG. 8

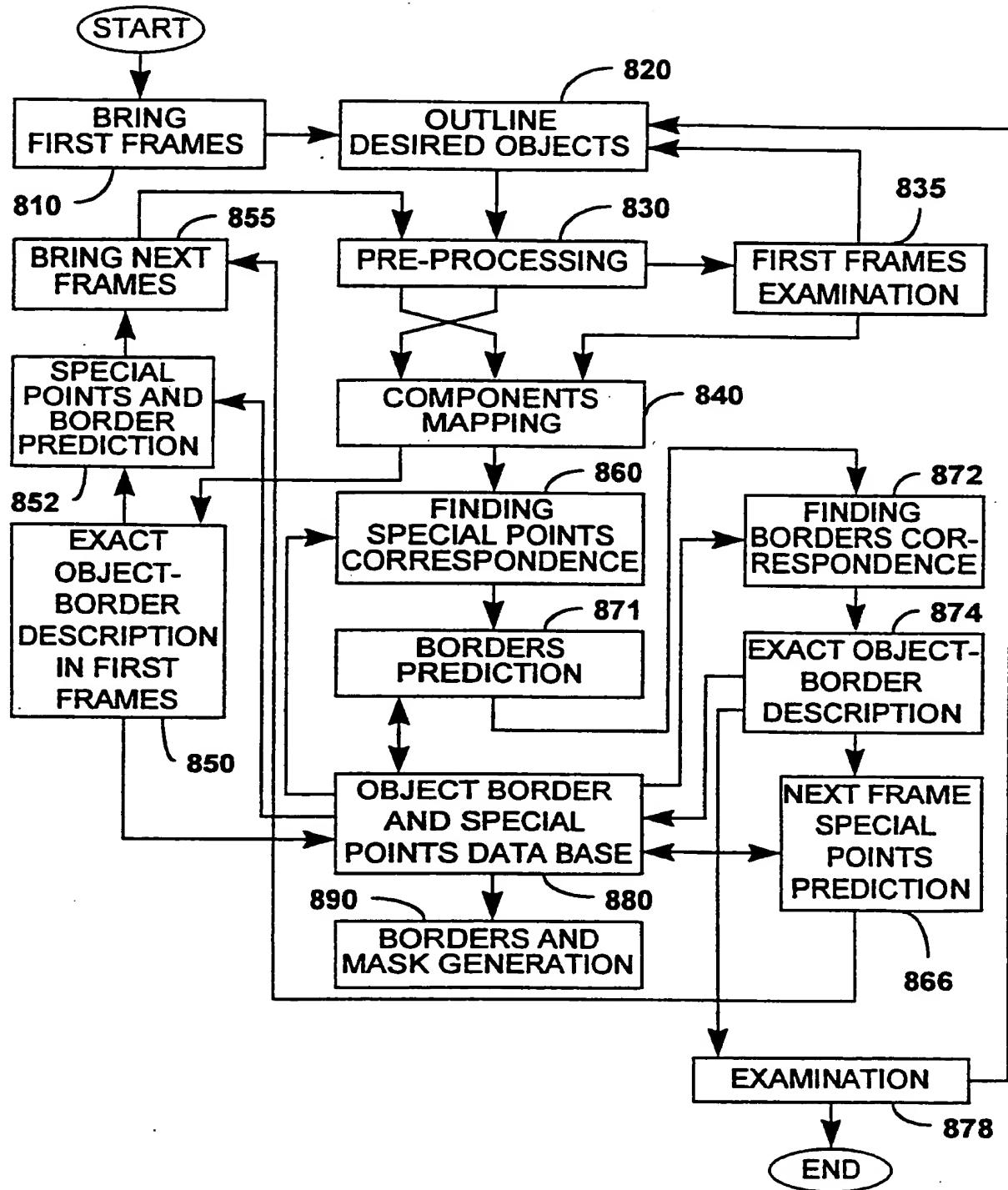
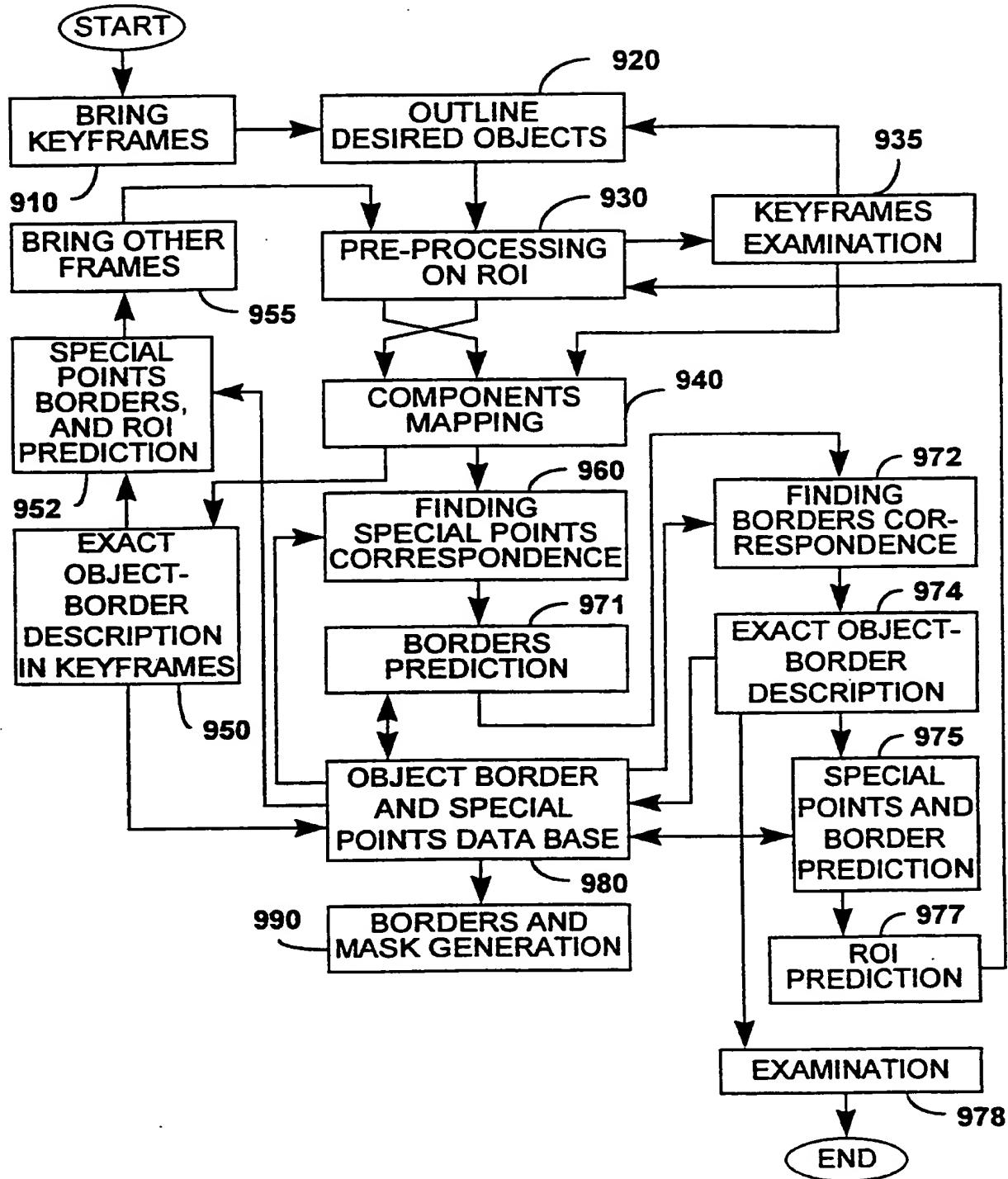


FIG. 9



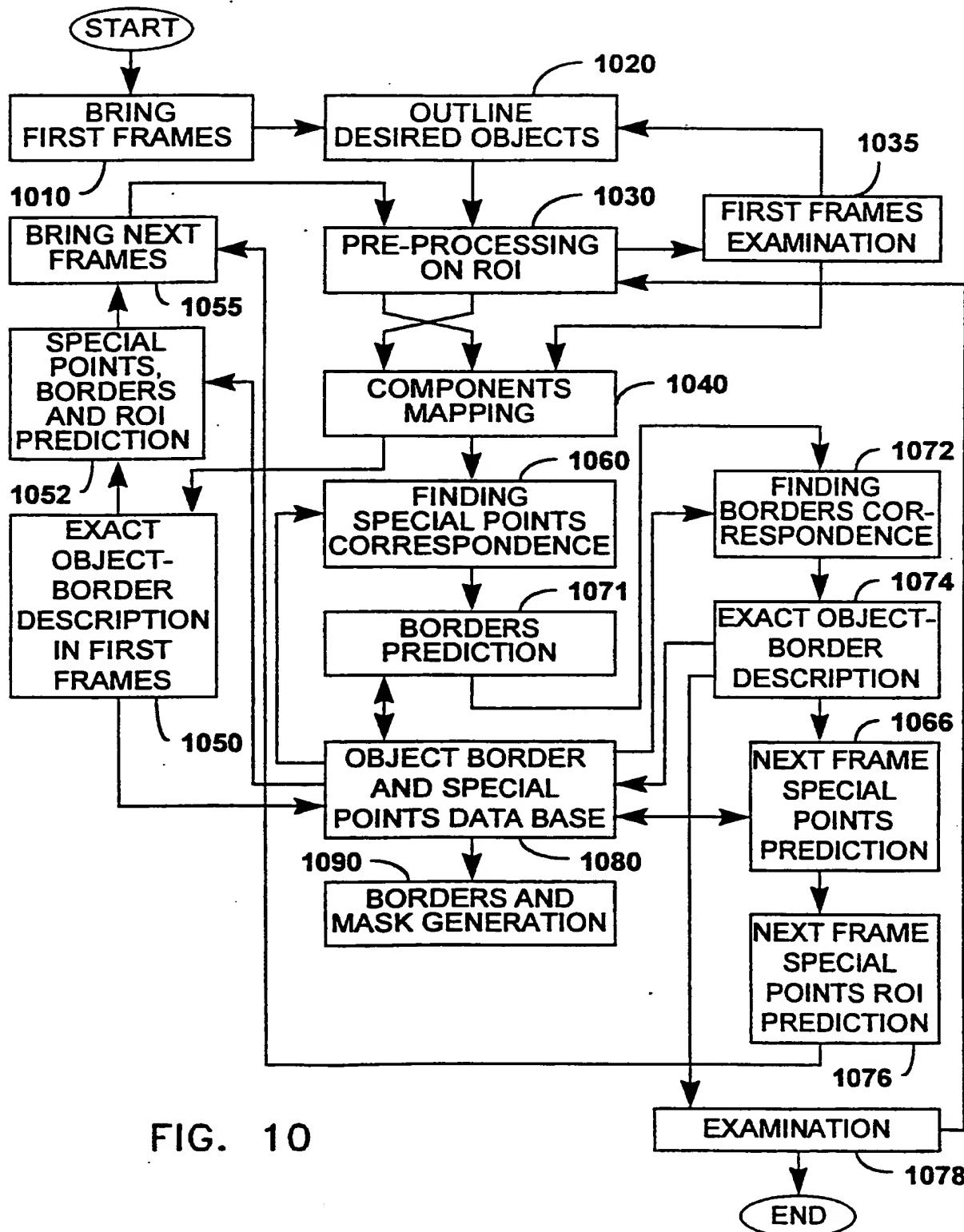


FIG. 10

FIG. 11

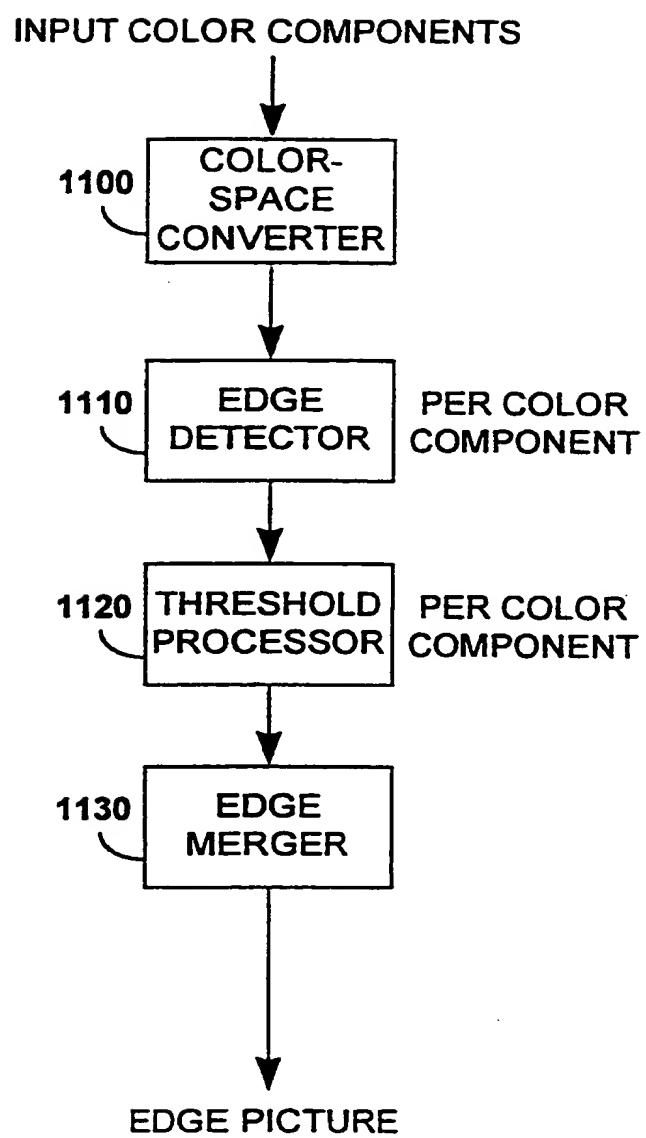
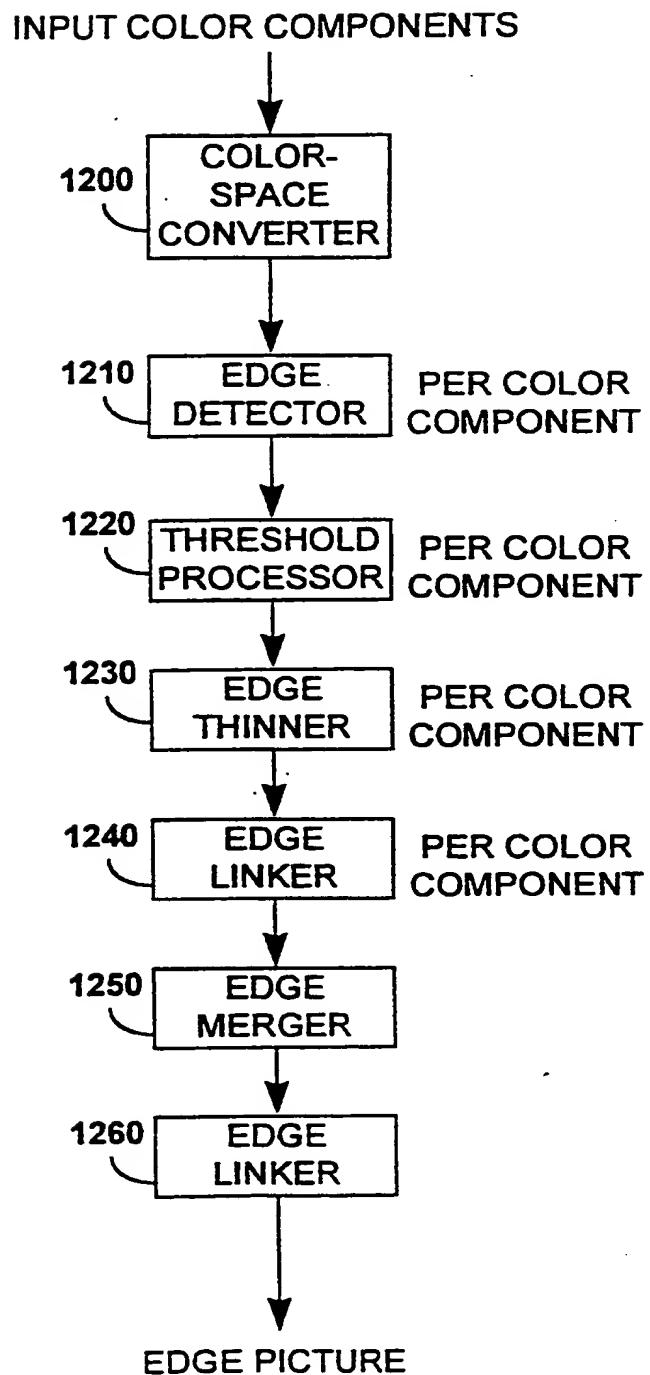


FIG. 12A

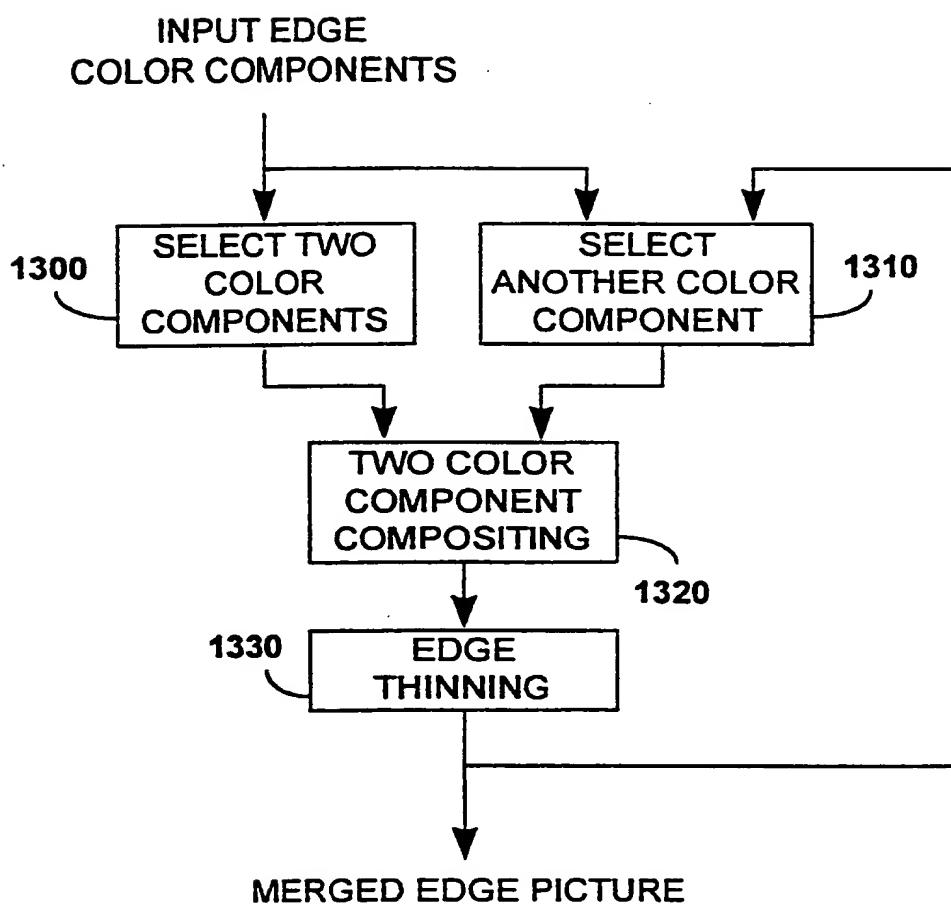


19 / 51

FIG. 12B

				1267	
		1266	1263	1267	
				1264	
				1264	
		1261	1262	1265	1265
	1261				
1261					

FIG. 13A



21/51

FIG. 13B

I3			I1		
	I3		I1	I2	I2
	I3	I2	I1	I3	I3
	I3	I2	I1		
		I3	I1	I3	I3
	I3	I1			
I3		I1	I2	I2	

22/51
FIG. 13C

I3						
	I3				I2	I2
	I3	I2			I3	I3
	I3	I2				
		I3			I3	I3
	I3					
I3			I2	I2		

FIG. 13D

I3						
	I3				I2	I2
		I2				
		I2				
		I3			I3	I3
	I3					
I3			I2	I2		

23/51

FIG. 13E

I3			I1		
	I3		I1	I2	I2
		I2	I1		
		I2	I1		
		I3	I1	I3	I3
	I3	I1			
I3		I1	I2	I2	

FIG. 13F

I3			I1		
	I3		I1	I2	I2
		I2	I1		
			I1		
			I1	I3	I3
	I3	I1			
I3		I1	I2	I2	

24/51

FIG. 14A

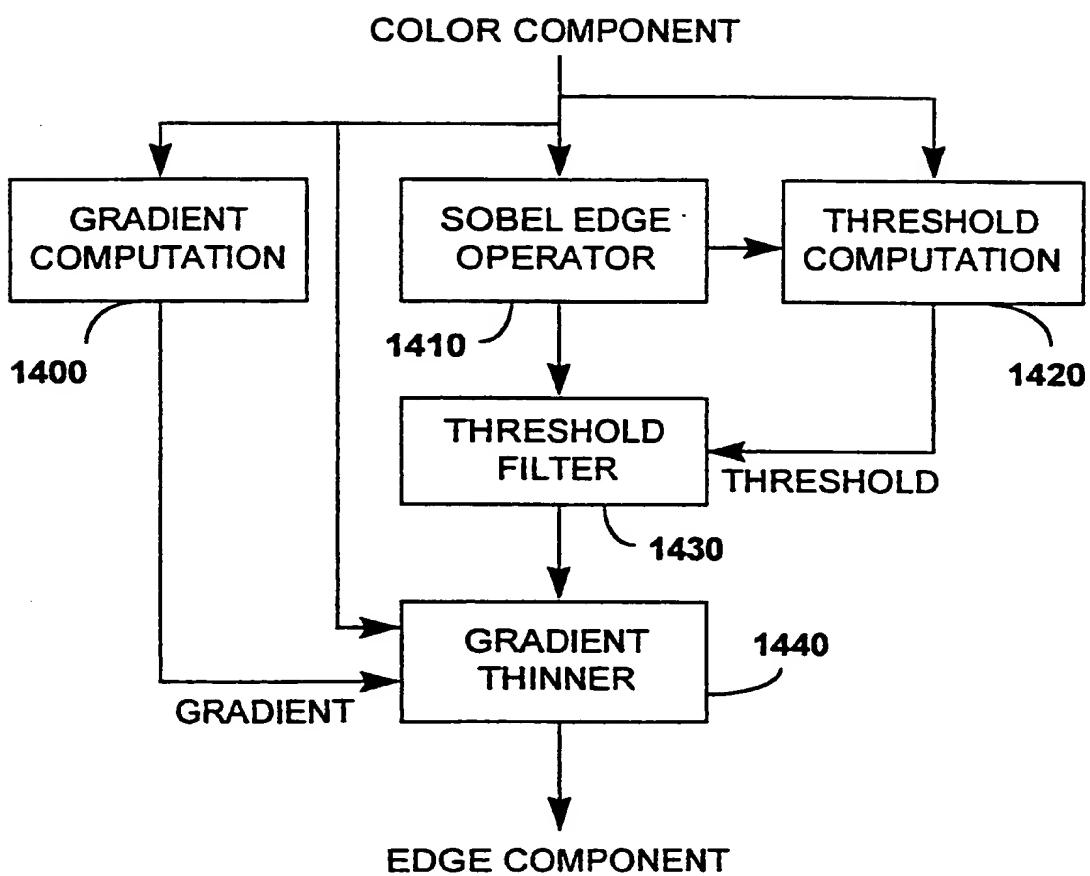
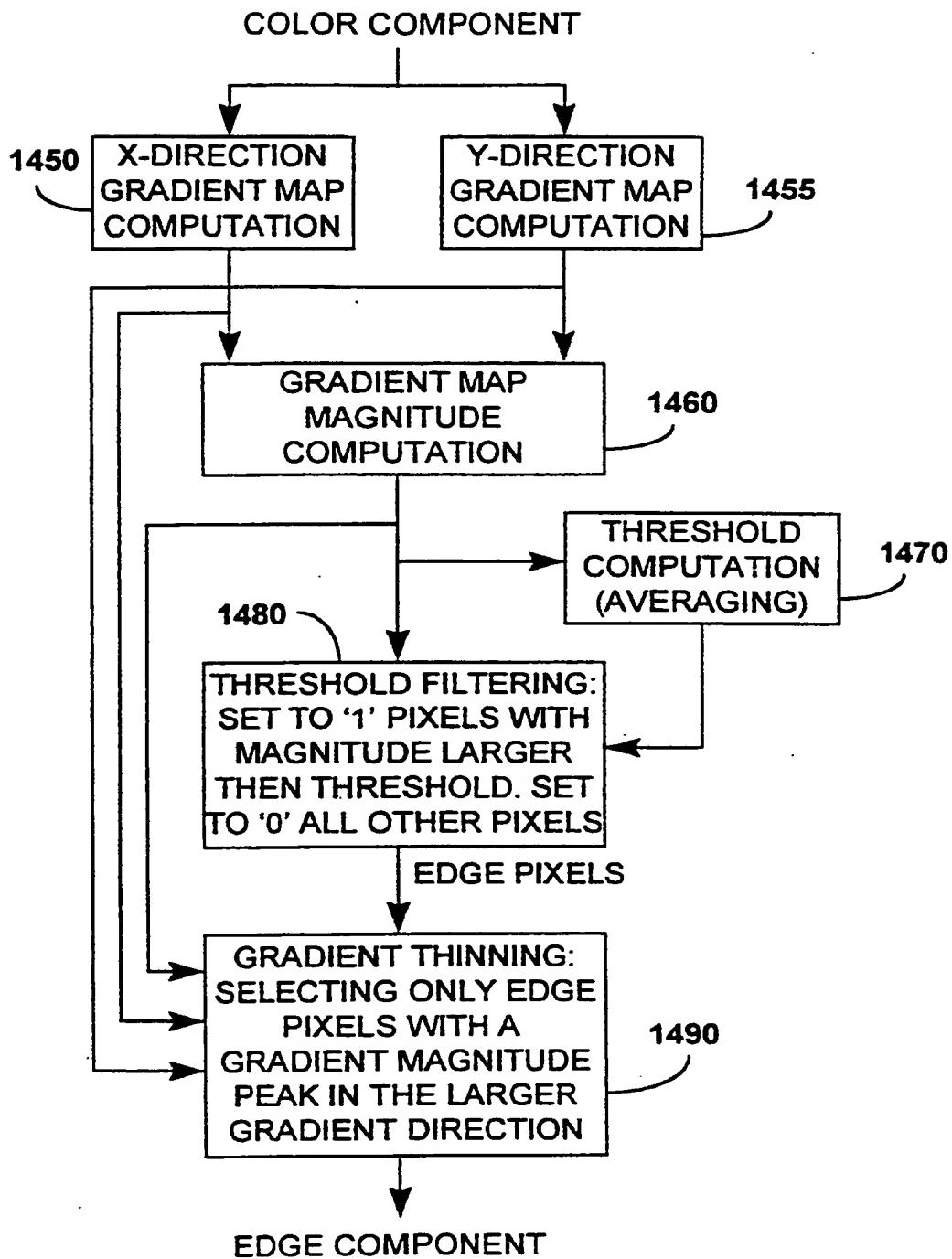
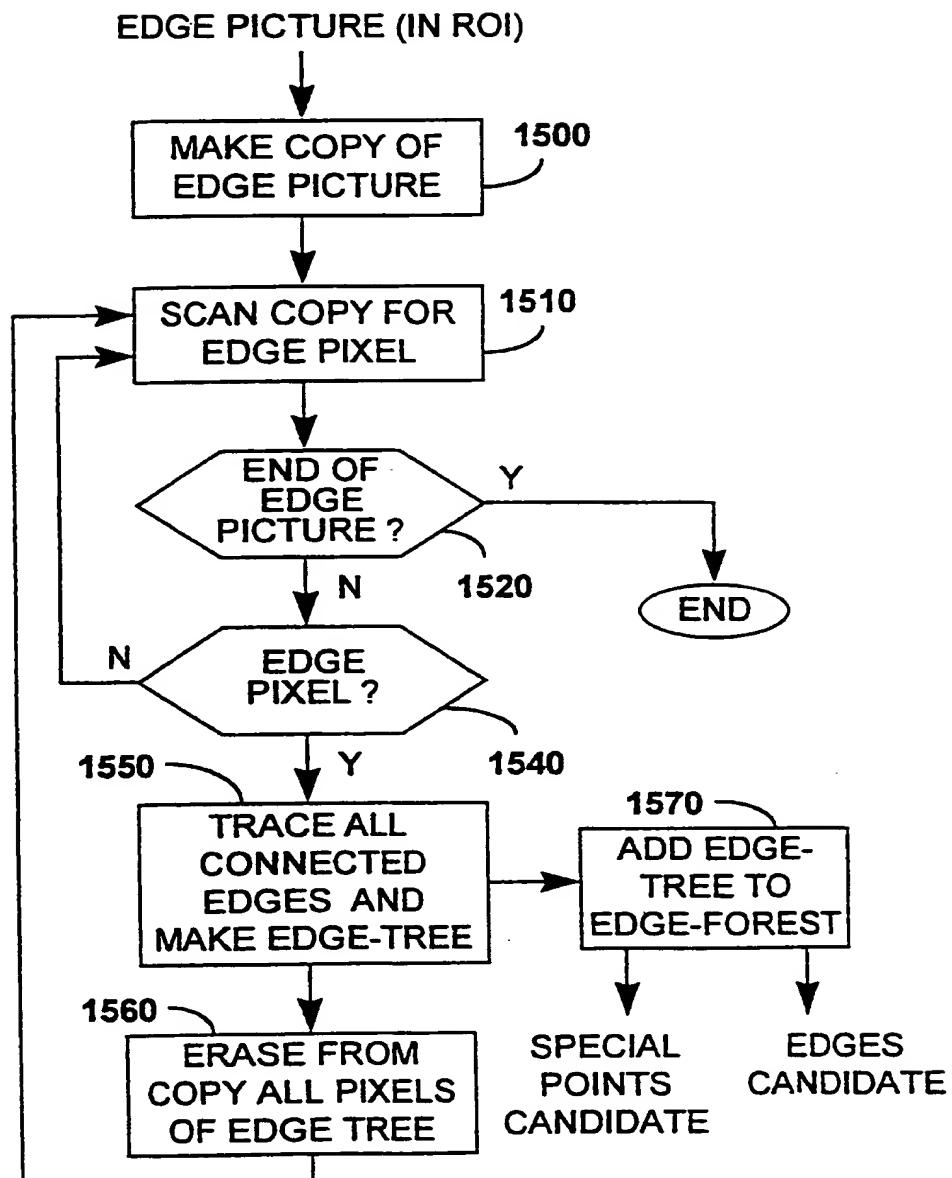


FIG. 14B



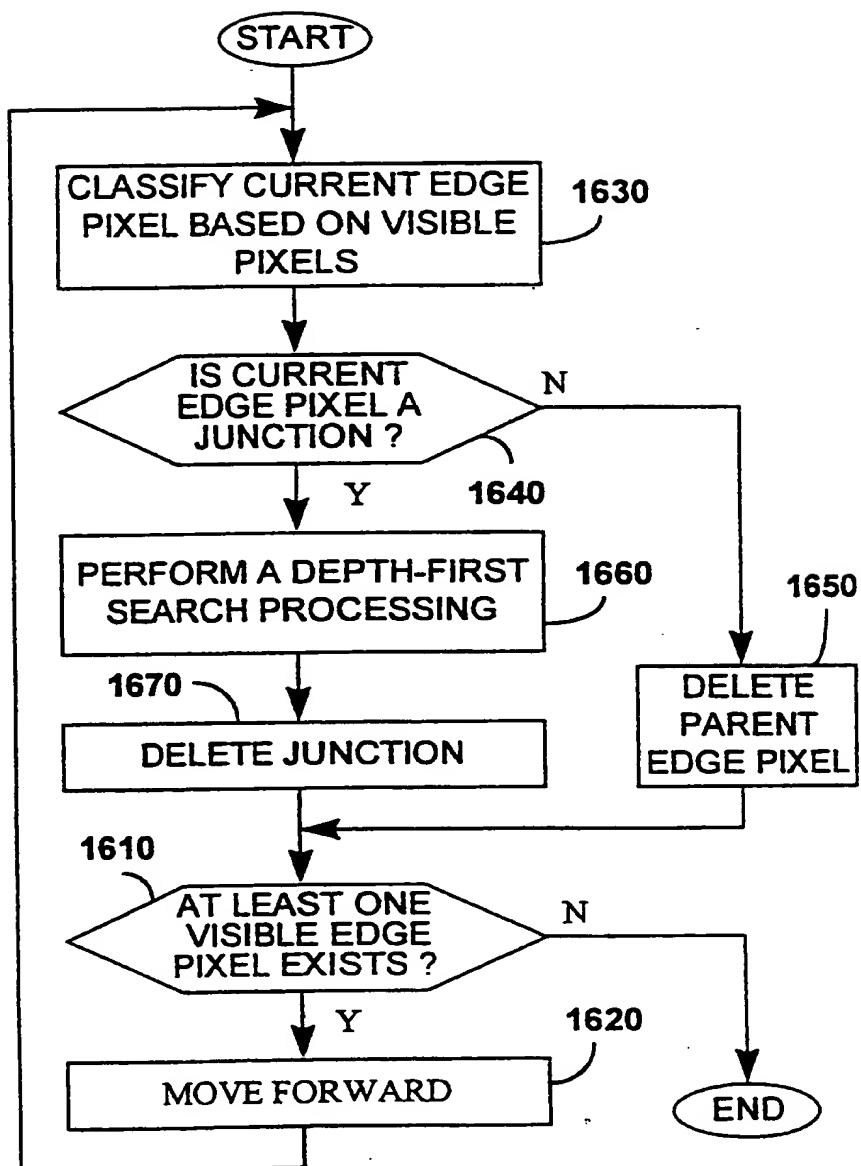
26/51

FIG. 15

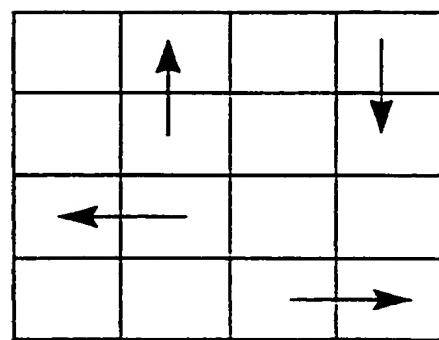
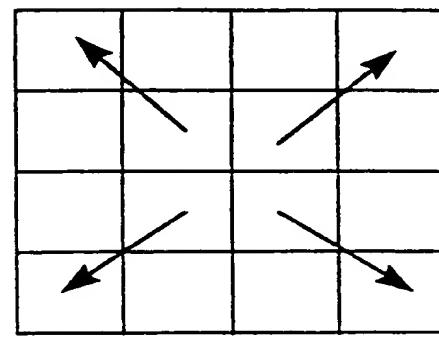


27/51

FIG. 16A



28/51

FIG. 16B**FIG. 16C**

29/51

FIG. 16D

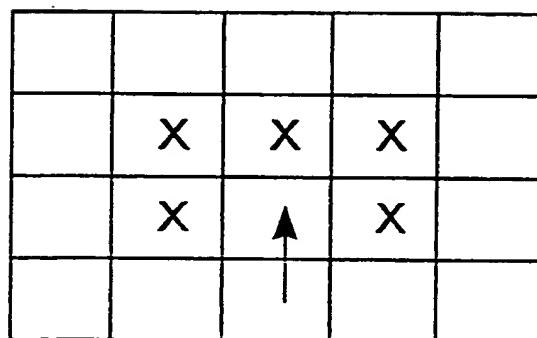
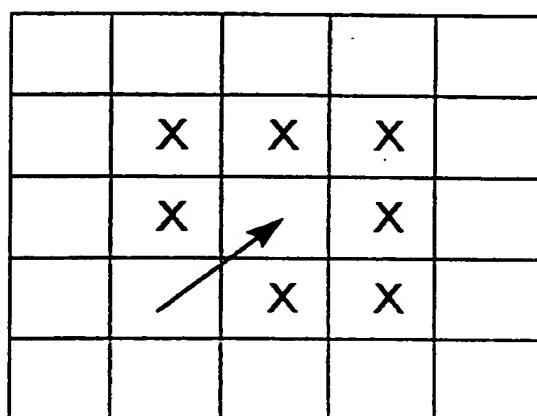


FIG. 16E



30/51

FIG. 16F

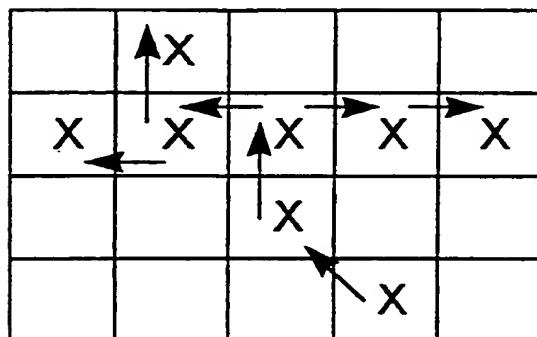


FIG. 16G

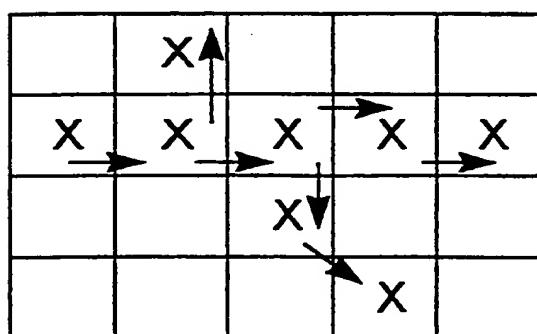
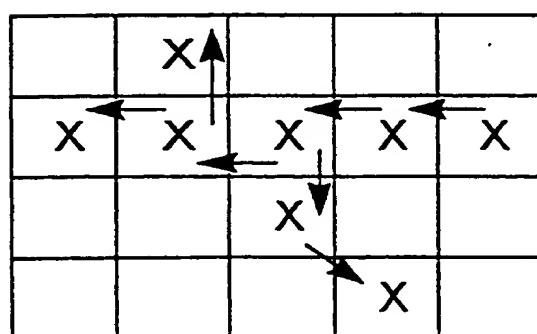
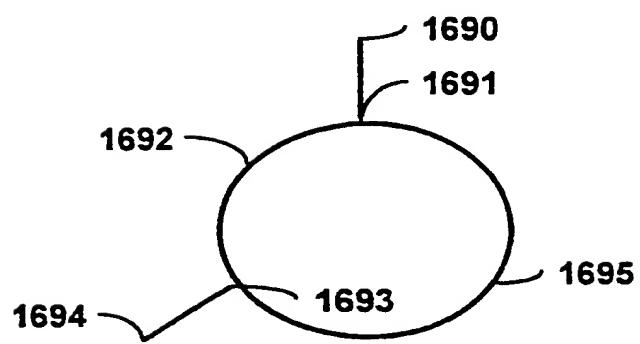


FIG. 16H



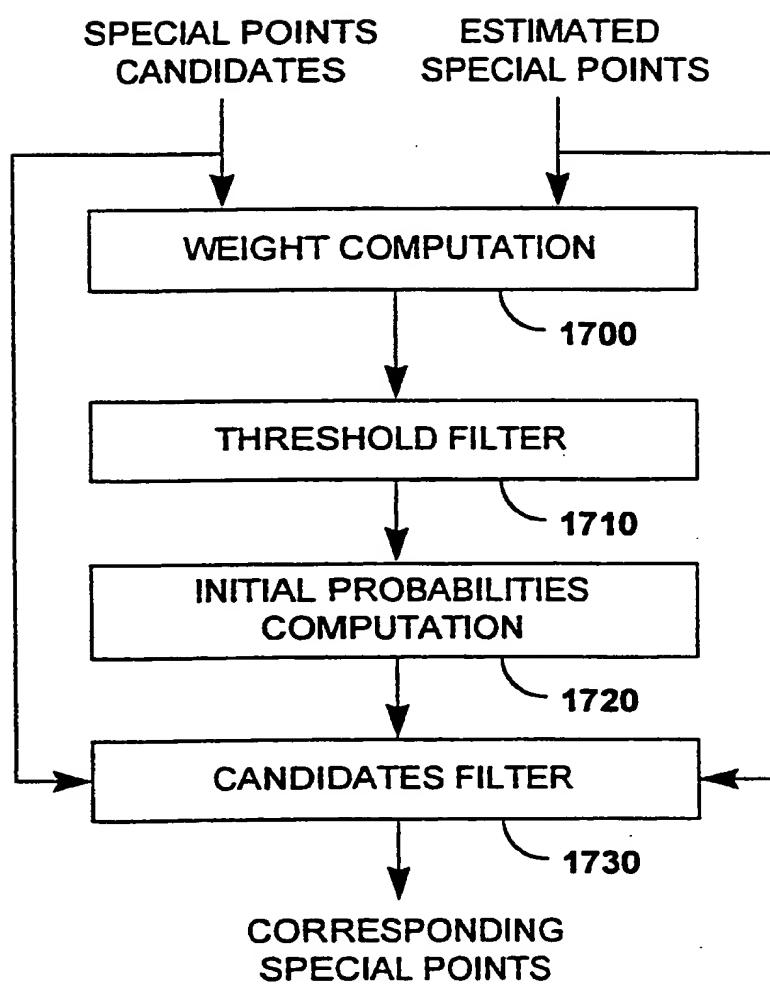
31/51

FIG. 16I



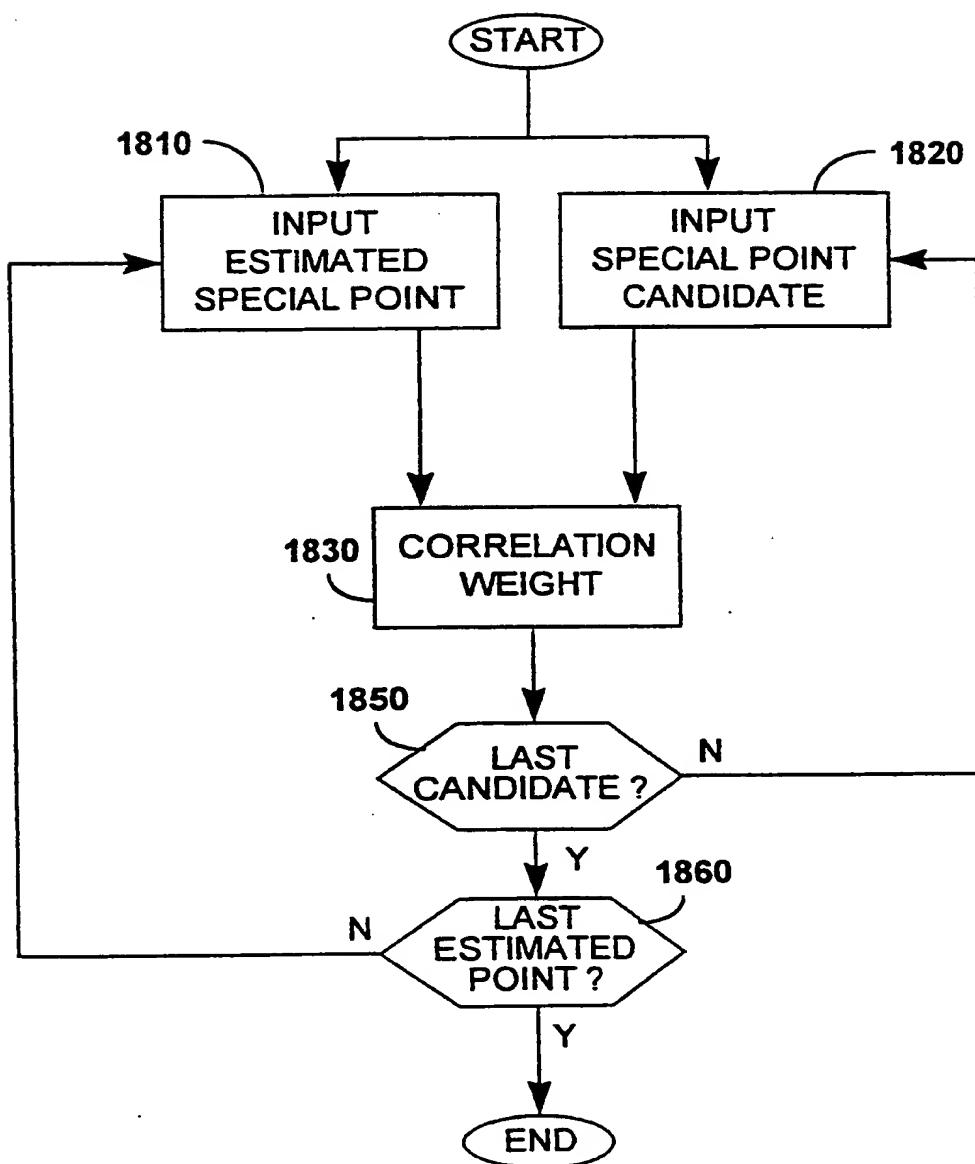
32/51

FIG. 17



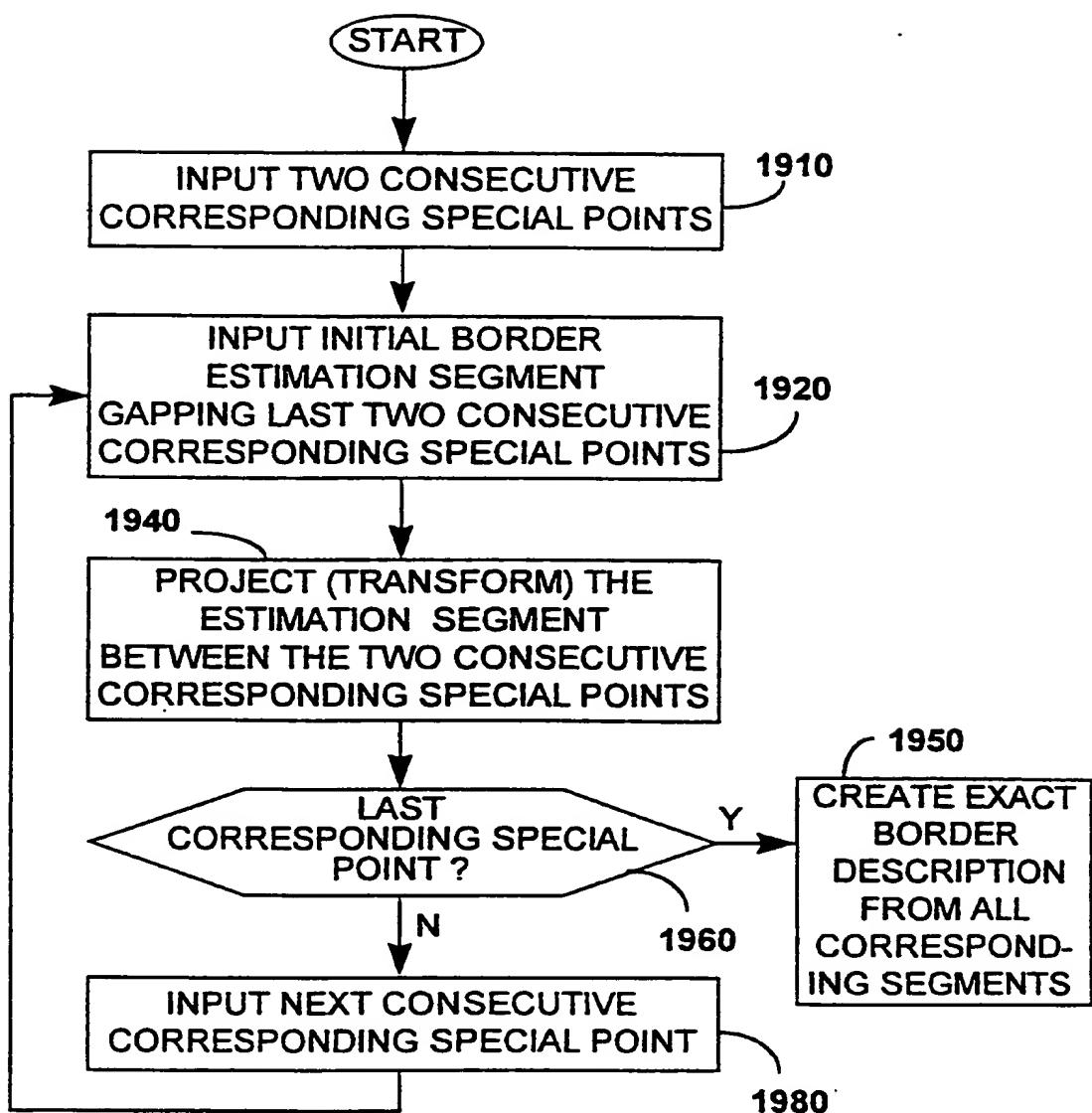
33/51

FIG. 18



34/51

FIG. 19



35/51

FIG. 20

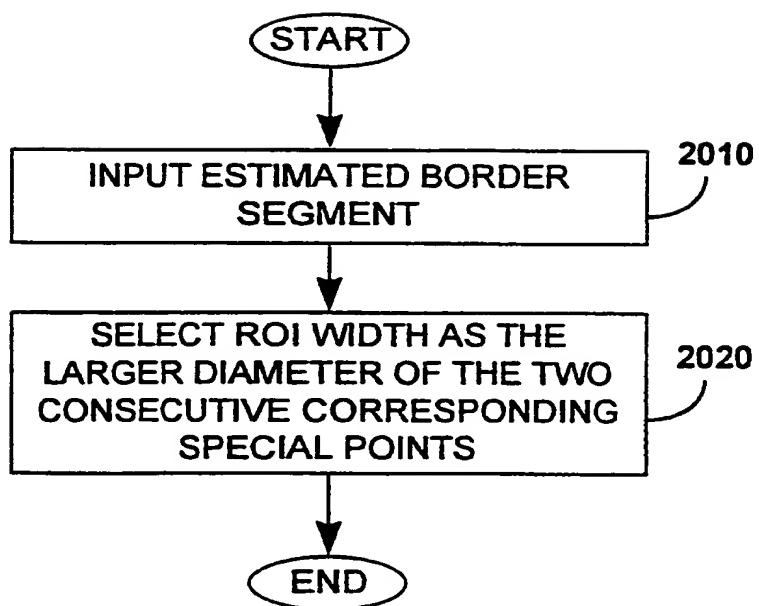
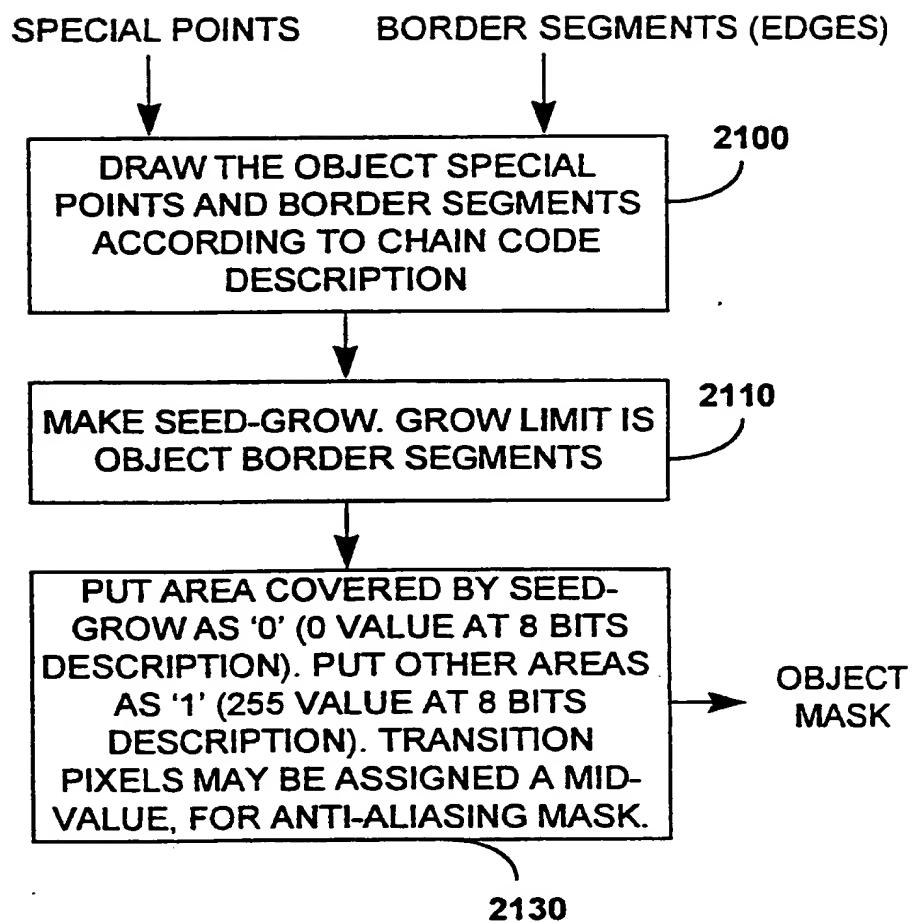
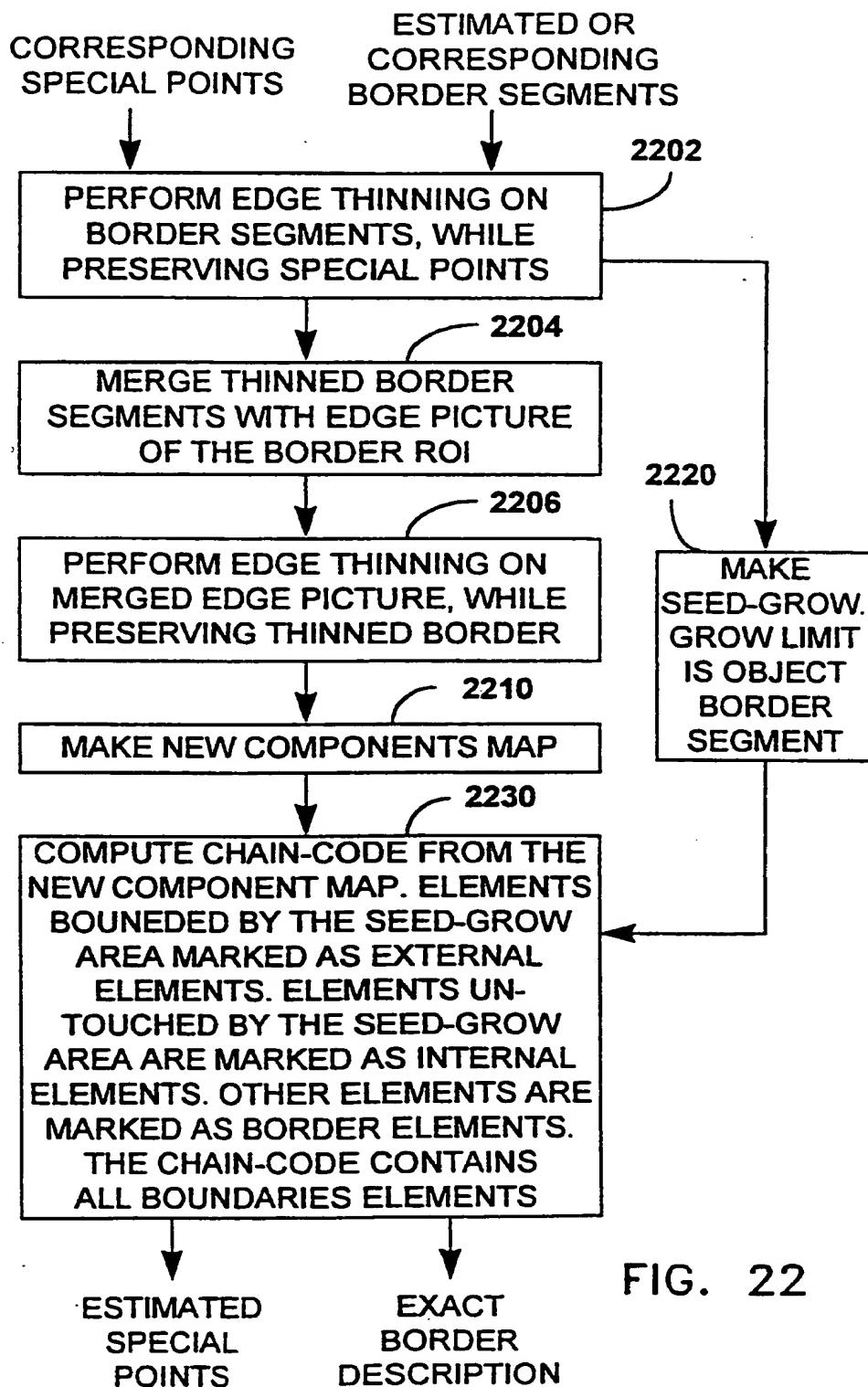
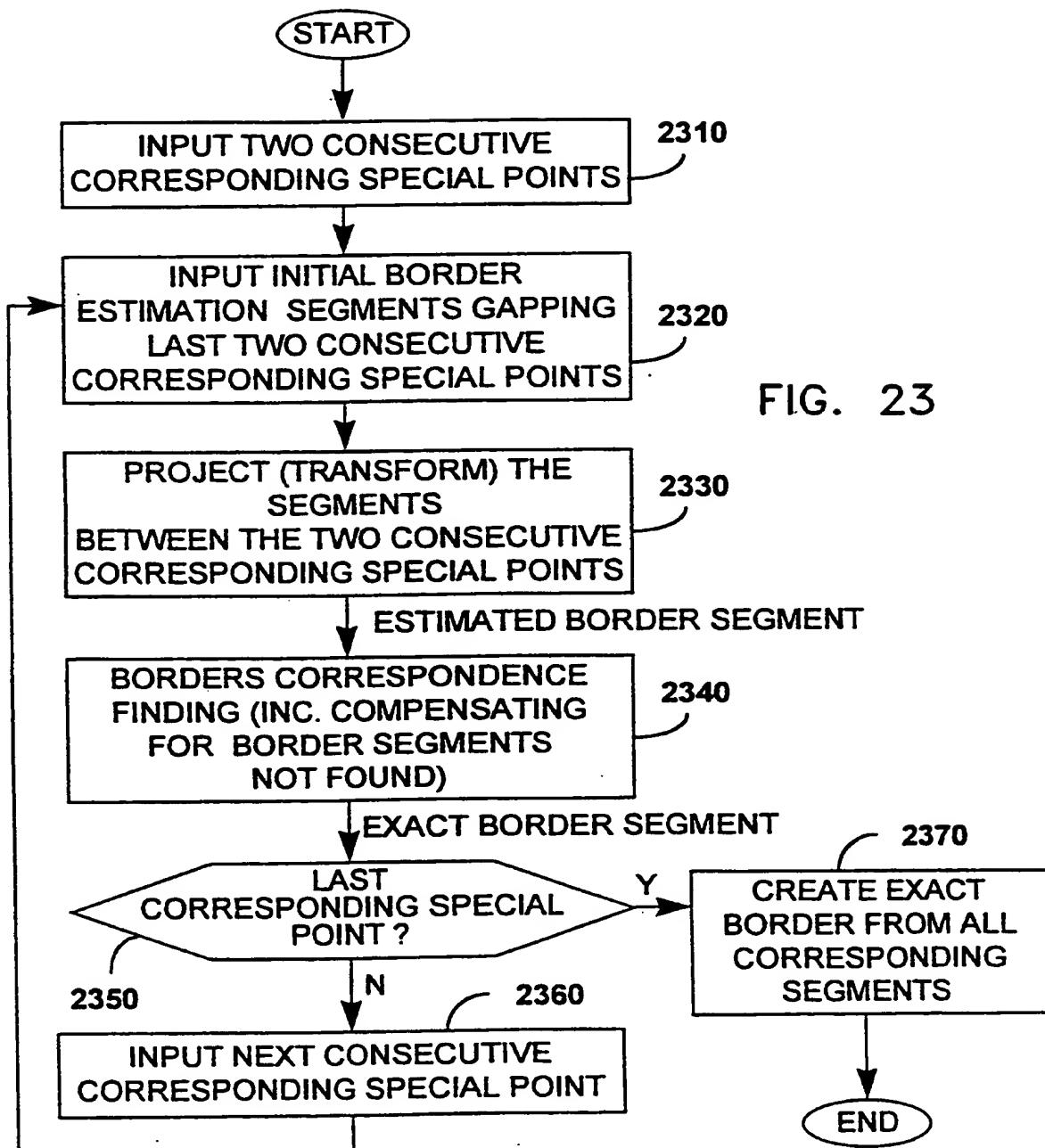


FIG. 21



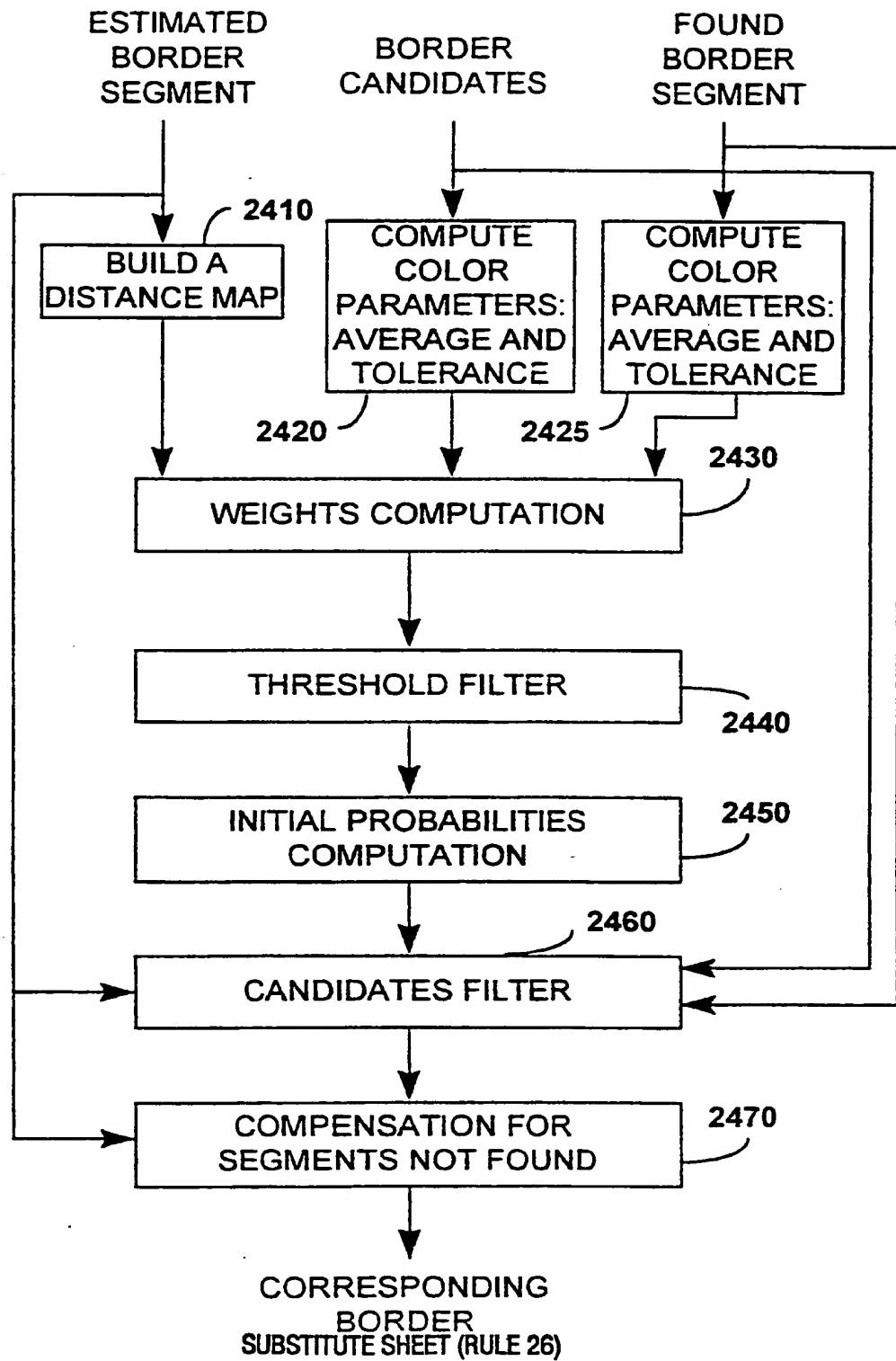
37/51

**FIG. 22**



39/51

FIG. 24



40/51

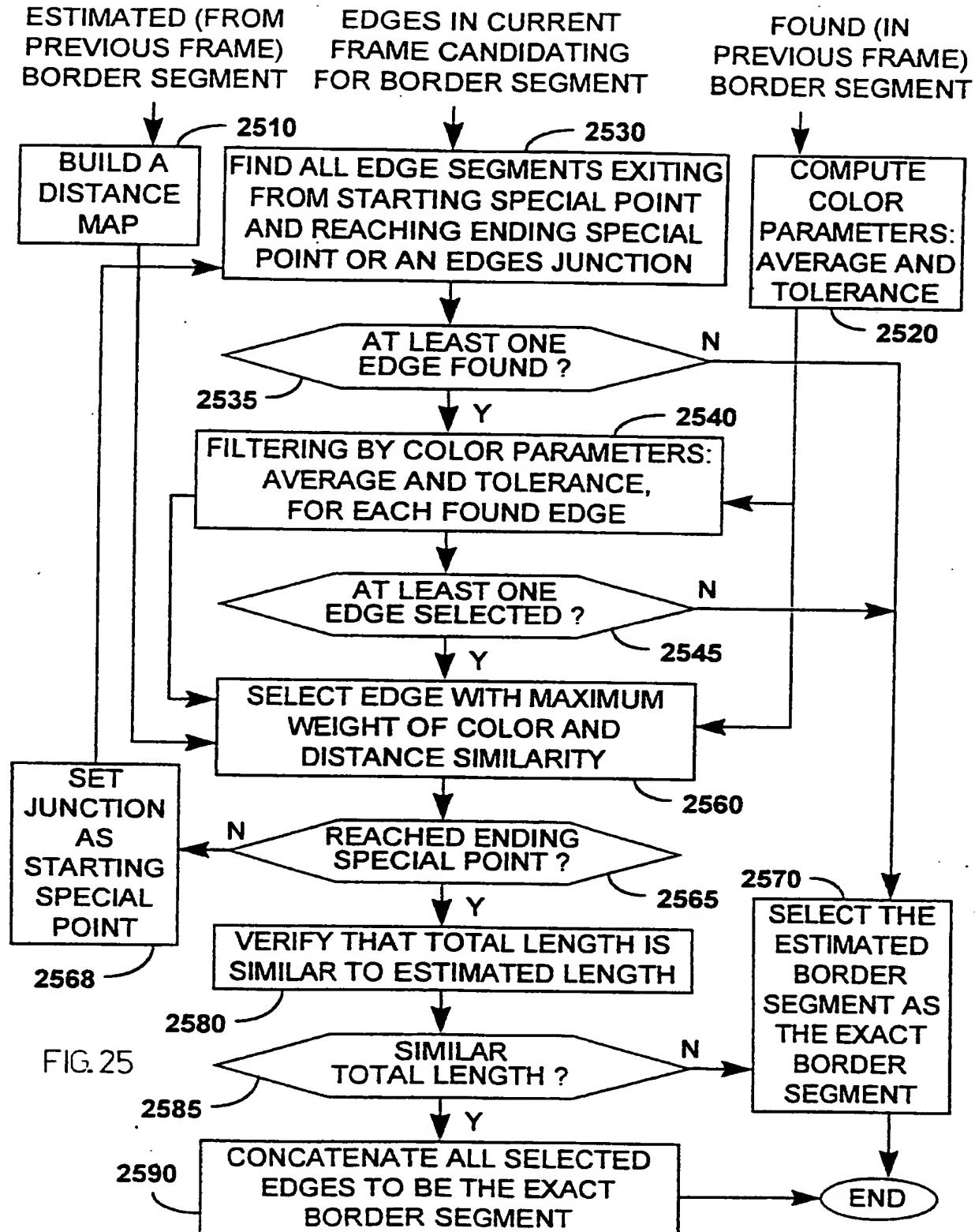
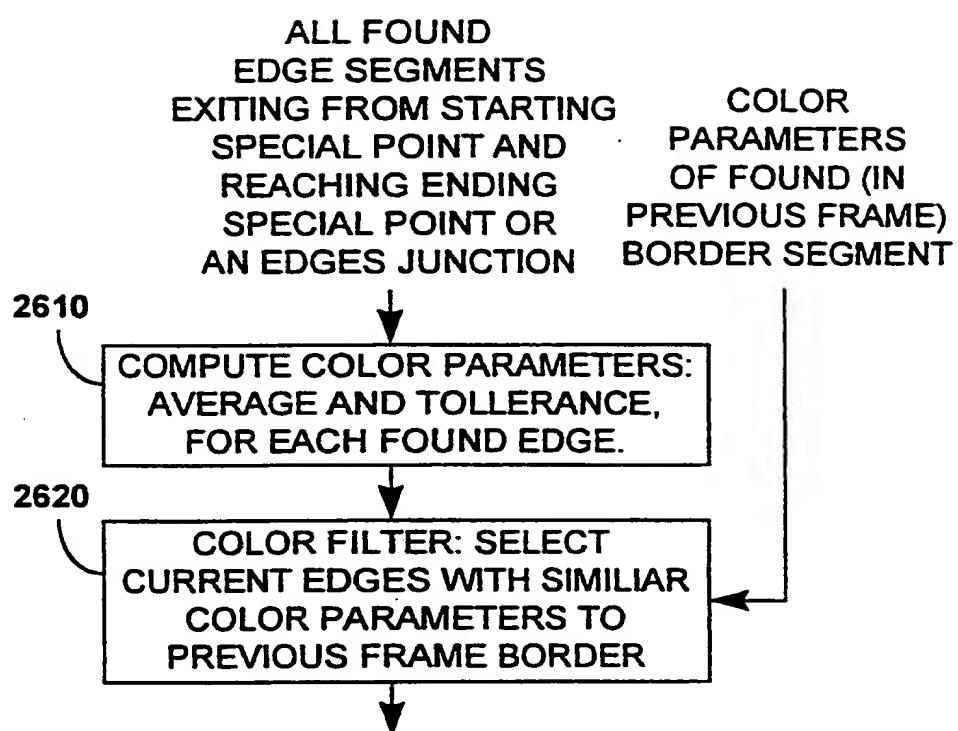


FIG. 26



42/51

FIG. 27

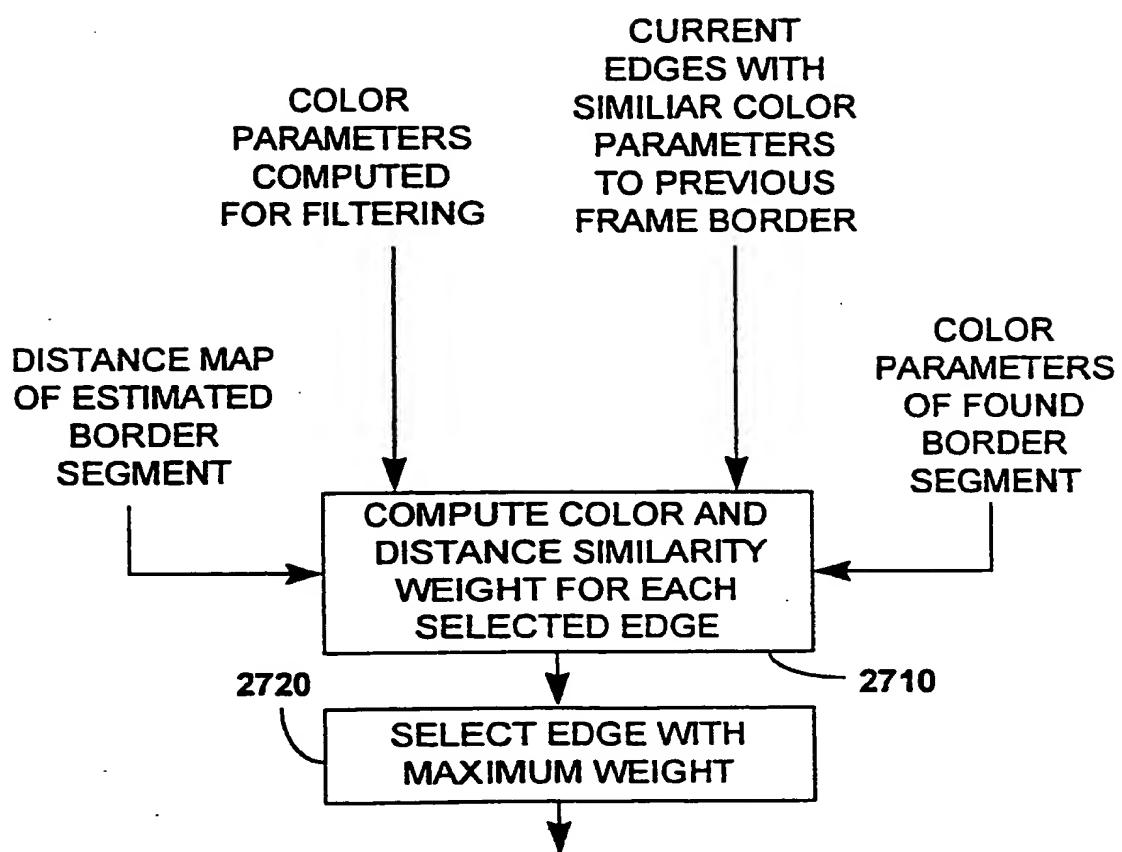
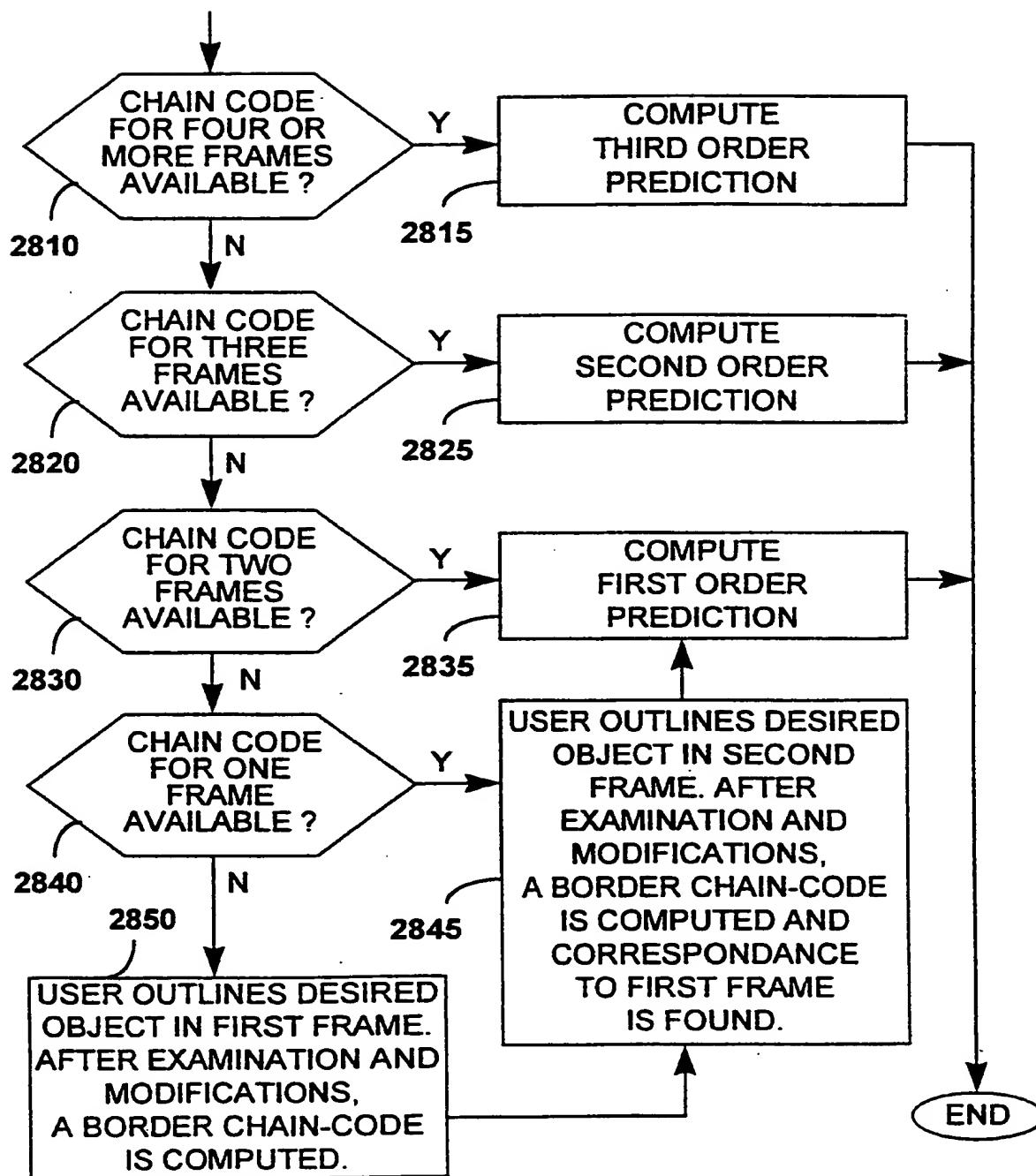
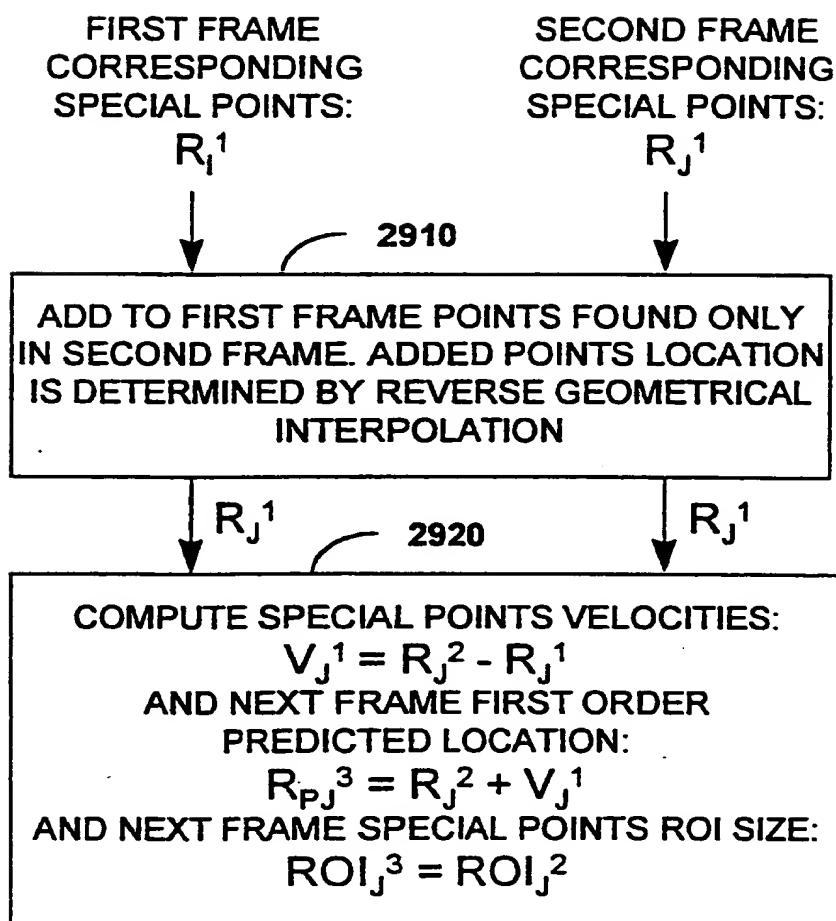


FIG. 28



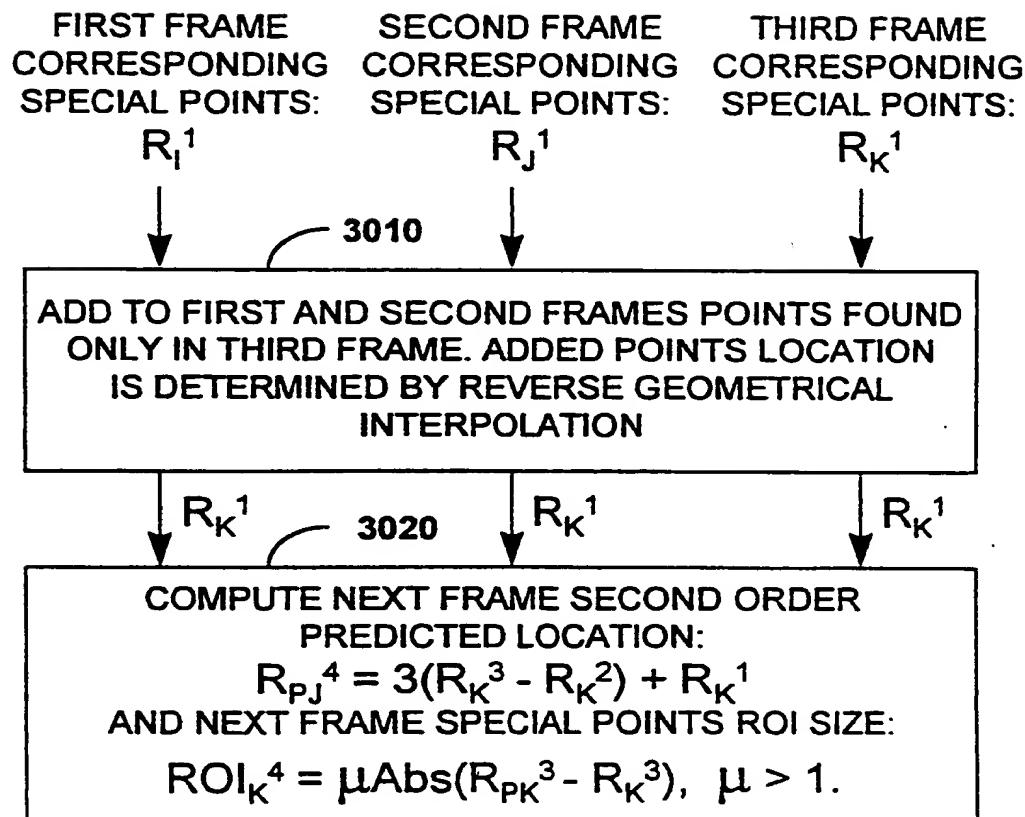
44/51

FIG. 29



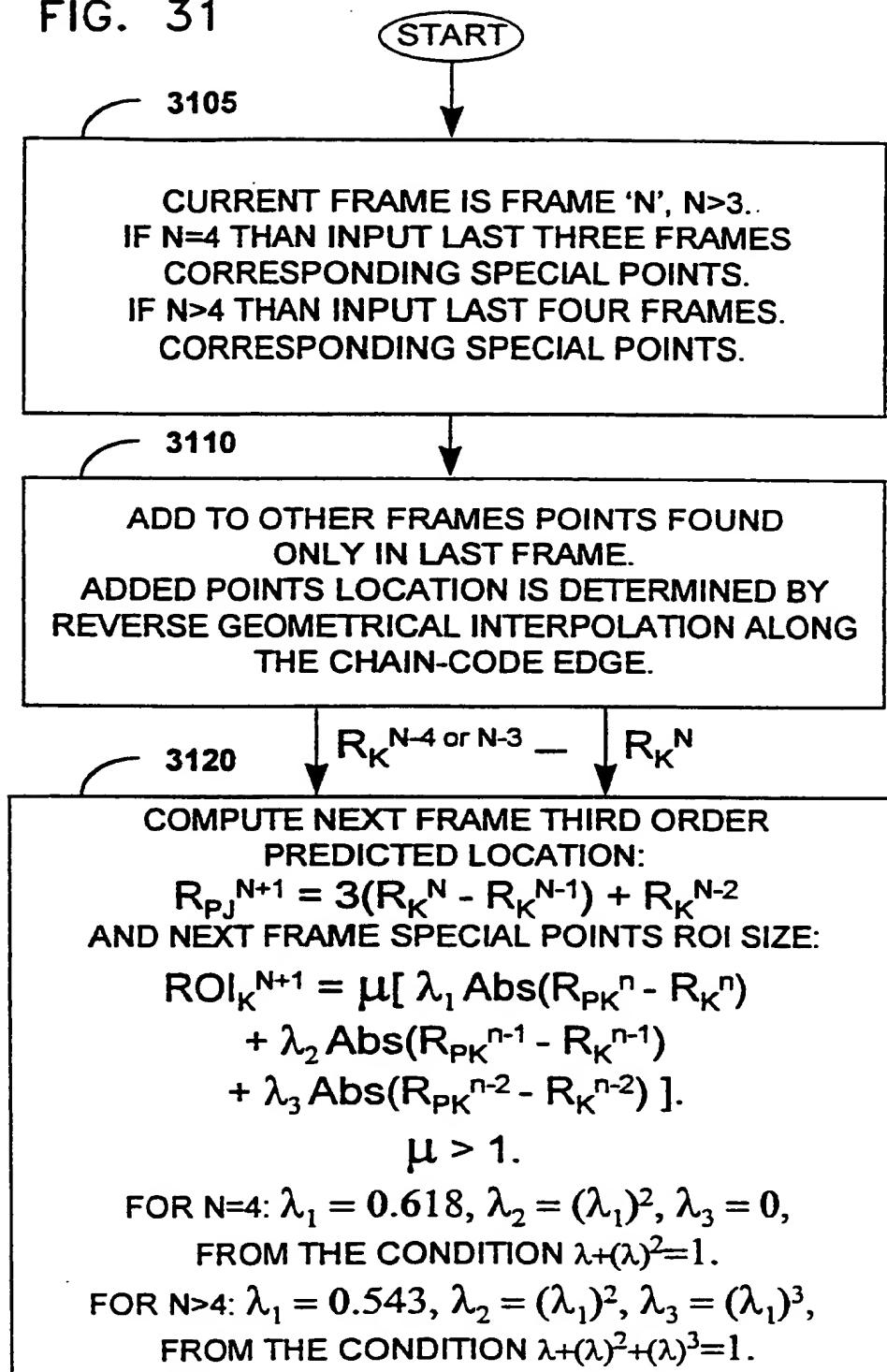
45/51.

FIG. 30



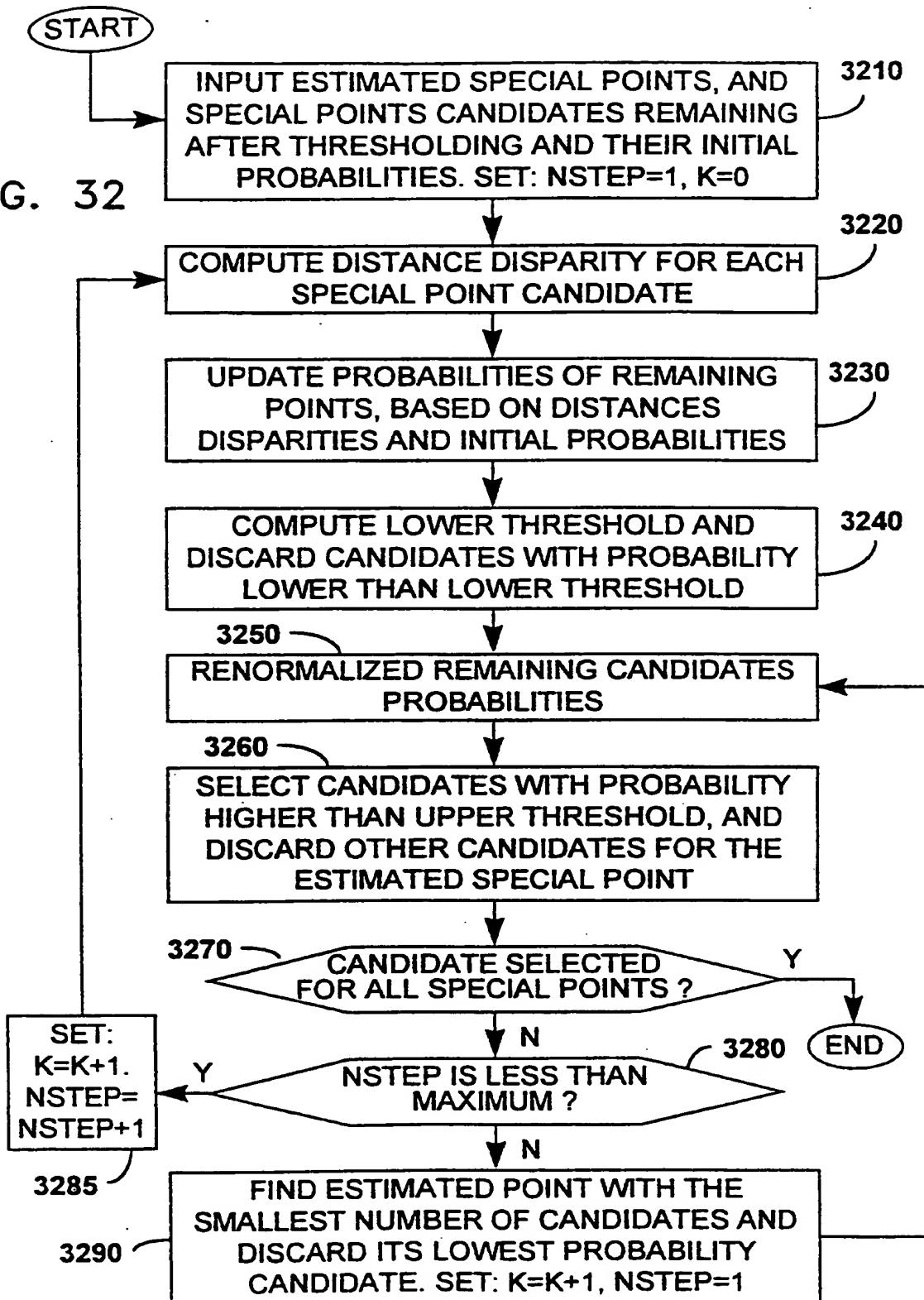
46/51

FIG. 31



47/51

FIG. 32



48/51

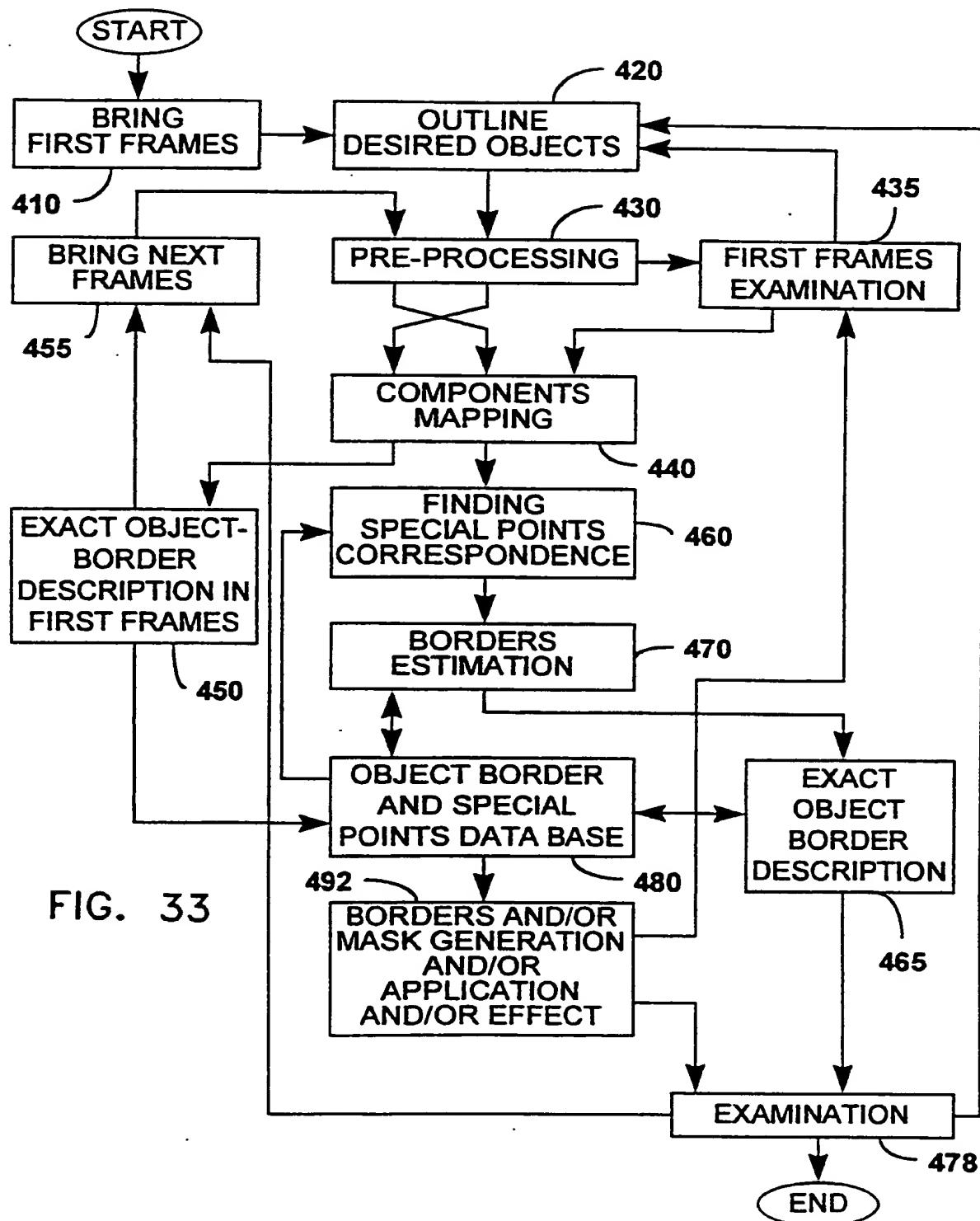


FIG. 33

49/51

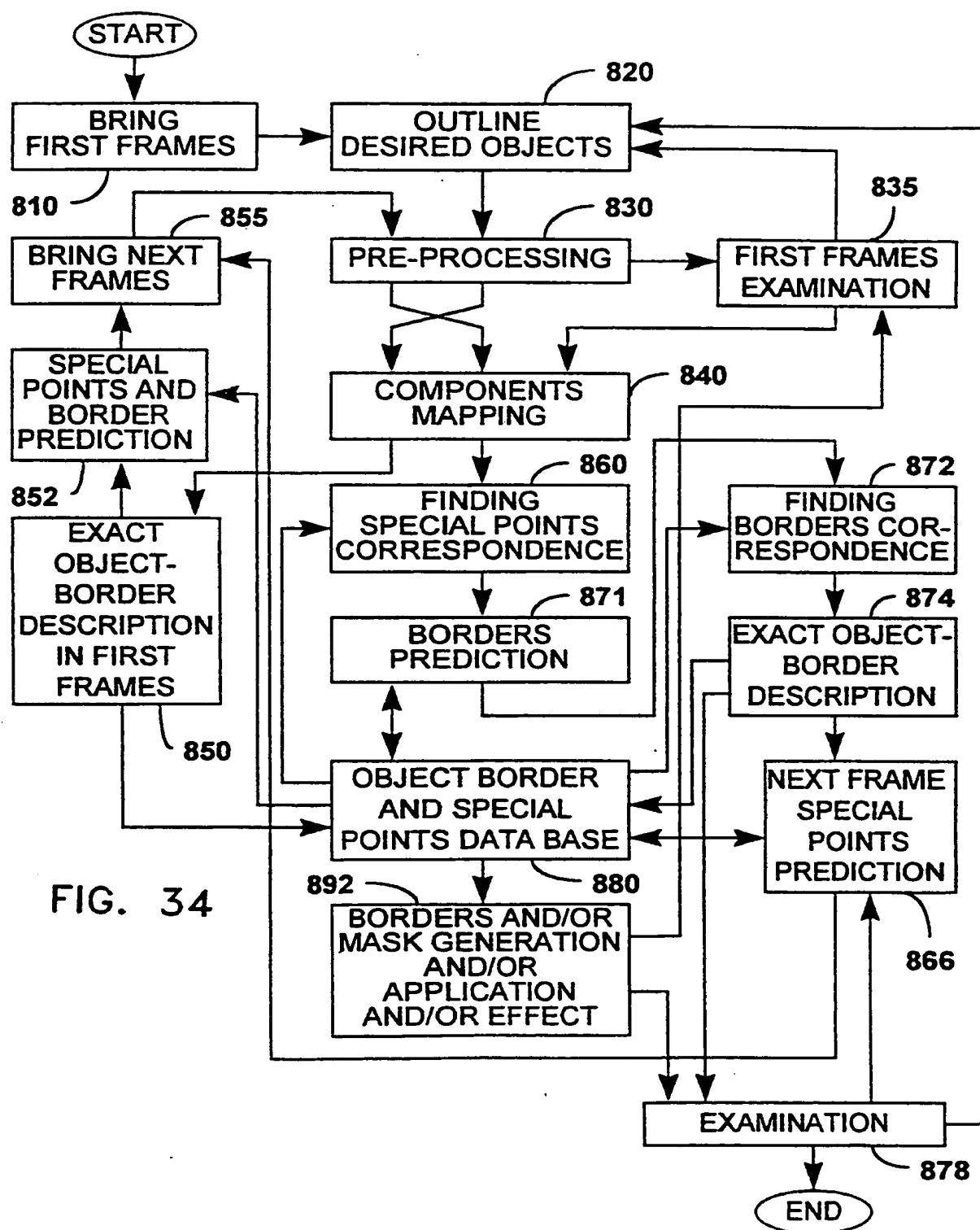


FIG. 34

50/51

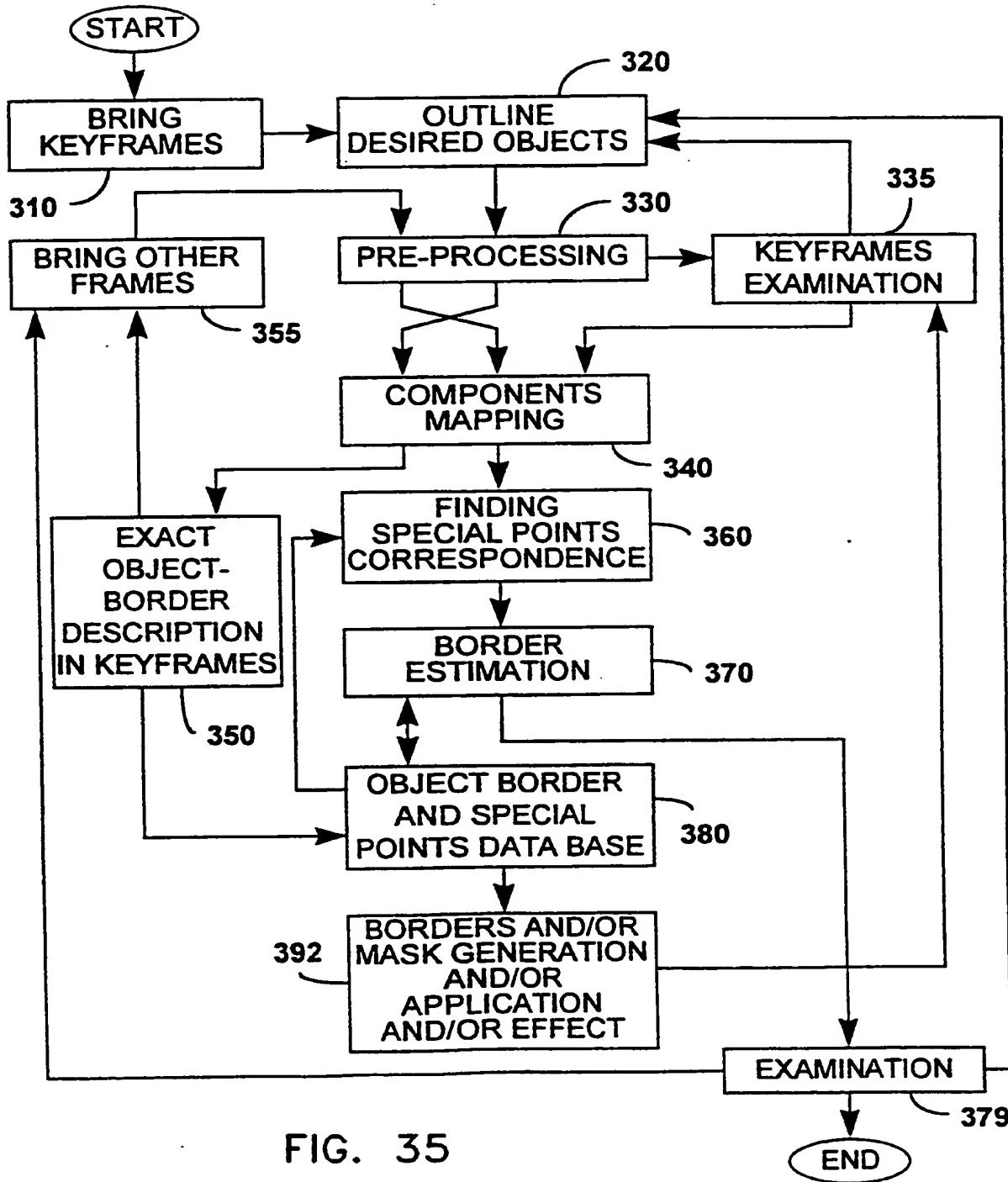
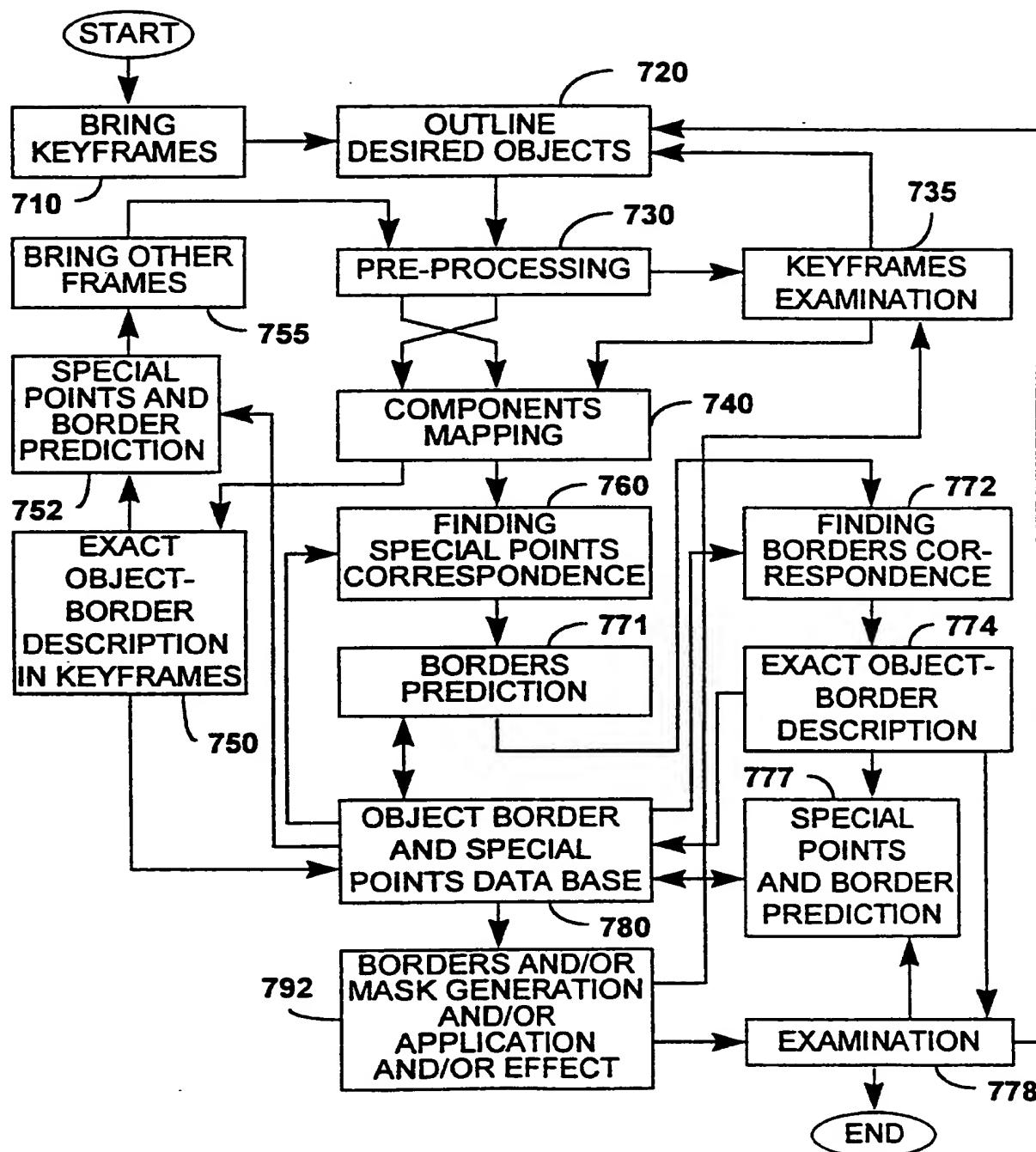


FIG. 35

FIG. 36



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.